Copy No._____

Defence Research and Development Canada
Recherche et développement pour la défense Canada

DEFENCE R&D DÉFENSE

Technical Preparation for an Initial Maritime Virtual Battle Experiment

Behind the Scenes of VBE CA-1

Tania E. Wentzell

Garfield R. Mellema

Defence R&D Canada – Atlantic

Technical Memorandum

DRDC Atlantic TM 2004-014

July 2004

Canada

105-04516/

Technical Preparation for an Initial Maritime Virtual Battle Experiment

Behind the Scenes of VBE CA-1

Tania E. Wentzell
Garfield R. Mellema

Defence R&D Canada – Atlantic

Technical Memorandum
DRDC Atlantic TM 2004-014
July 2004

Authors

Tania E. Wentzell and Garfield R. Mellema

Approved by

J. S. Kennedy

Section Head / Maritime Information and Combat Systems

Approved for release by

Kirk Foster

DRP Chair

© Her Majesty the Queen as represented by the Minister of National Defence, 2004

© Sa majesté la reine, représentée par le ministre de la Défense nationale, 2004

Abstract

The Virtual Combat Systems (VCS) Group at Defence Research and Development Canada – Atlantic (DRDC Atlantic) is aiming to provide a combat systems experimentation facility to assist with the research programs of scientists within the Maritime Information and Combat Systems (MICS) section and beyond. The first such experiment, known as Virtual Battle Experiment Canada-1 (VBE CA-1), investigated the influence of coalition sharing of broadband passive sonar track segment data.

The experiment required sonar operators to monitor vessel traffic in a strait and make decisions about which sonar tracks were from common targets. In some cases, sonar track data was available to the operator from a coalition ship, and in other cases the operators were only provided with sonar track data from their own sonar. The experiment was run multiple times with various subjects to obtain statistically relevant results.

This paper focuses on the technical activities that were completed in preparation for this experiment. Scenario development, selection, development and integration of simulation components, and testing of the architecture are discussed.

Résumé

Le Groupe des systèmes de combat virtuel (SCV) de Recherche et développement pour la défense Canada – Atlantique (RDDC Atlantique) vise à fournir une installation de recherche sur les systèmes de combat aux scientifiques de la section d'information maritime et de systèmes de combat (MICS) et d'autres sections pour les aider à mettre en œuvre leurs programmes de recherche. La première expérience du genre, connue aussi sous le nom de première expérience de combat virtuel du Canada-1 (VBE CA-1), porte sur l'incidence du partage, au sein de la coalition, des données sur les segments de trajectoire provenant de sonars passifs large bande.

Durant l'expérience, les opérateurs de sonar devaient surveiller le trafic des navires dans un détroit et identifier les trajectoires sonar qui provenaient de cibles communes. Dans certains cas, les données sur les trajectoires sonar étaient fournies aux opérateurs par un navire de la coalition; dans d'autres cas, les opérateurs n'avaient accès qu'aux données de trajectoires sonar provenant de leur propre sonar. L'expérience a été répétée plusieurs fois avec divers sujets, afin d'obtenir des résultats pertinents du point de vue statistique.

Le présent document traite principalement des activités techniques qui ont été complétées pour la préparation de cette expérience. Il porte sur l'élaboration de scénarios ainsi que sur la sélection, le développement et l'intégration d'éléments de simulation et sur l'essai de l'architecture.

Executive Summary

Introduction

The Virtual Combat Systems (VCS) Group at Defence Research and Development Canada – Atlantic (DRDC Atlantic) is aiming to provide a combat system experimentation facility to assist with the research programs of scientists within the Maritime Information and Combat Systems (MICS) section and beyond. The first such experiment, known as Virtual Battle Experiment Canada-1 (VBE CA-1), investigated the influence of sharing coalition broadband passive sonar track segment data.

The VBE CA-1 infrastructure used the High Level Architecture (HLA) and consisted of a number of individual simulation components (federates), some which were obtained from the Defence Science and Technology Organisation (DSTO) in Australia, and some which were developed at DRDC Atlantic to satisfy the needs of this experiment.

VBE CA-1 considered the situational awareness benefits of coalition sharing of unprocessed, broadband passive sonar track segment data. Sonar operators were tasked to monitor vessel traffic in a strait and make decisions about which sonar tracks were from common targets. In some cases, sonar track data was available from the operator's ownship and from a second (coalition) ship, while in other cases the operators were provided only with sonar tracks from their own sonar. The experiment was run multiple times, with various subjects, to obtain statistically relevant results.

This paper focuses on the technical activities that were completed in preparation for this experiment. Scenario development, selection and integration of simulation components, and testing of the architecture will be discussed.

Results

The infrastructure obtained from DSTO was integrated with federates developed at DRDC Atlantic. Experimental processes required to plan and execute a Human-in-the-Loop (HIL) virtual experiment were tested and refined.

Overall, the capability to simulate a maritime environment, immerse operators into the virtual environment, and obtain statistically relevant results for matters of relevance to the military was successfully demonstrated.

Significance

This experiment was the first of its kind executed by the VCS group. As such, the experimental team underwent a significant amount of learning in completing the experiment (integrating simulation components into the architecture, developing new

components, creating various scenarios, etc.). With the resultant knowledge base, a greater understanding of the effort that must be applied for the successful completion of a VBE has been obtained. Future experiments will benefit from this experience through more efficient planning and execution.

Virtual experimentation shows potential for initial assessment of new technologies and operational concepts. It is lower cost and low risk, particularly in comparison to that of taking a frigate or task group out to sea. It could be used to identify glitches or errors prior to making the financial and resource commitment to do real-world experimentation. As the environment becomes more sophisticated, it may not only provide insight prior to sea trials, but also reduce the volume of sea trials required overall.

Future Plans

This experiment is the first in a series of experiments within the Coalition Sonar Track Sharing (CoaSTS) programme that will look at the sharing of coalition data at various levels of development. Future experiments in this series will provide the operators with more sophisticated analysis tools (triangulation and/or Target Motion Analysis (TMA)).

Wentzell, T.E. and Mellema, G.R. 2004. Technical Preparation for an Initial Maritime Virtual Battle Experiment. DRDC Atlantic TM 2004-014 Defence R&D Canada – Atlantic.

Sommaire

Introduction

Le Groupe des systèmes de combat virtuel (SCV) de Recherche et développement pour la défense Canada – Atlantique (RDDC Atlantique) vise à fournir une installation de recherche sur les systèmes de combat aux scientifiques de la section d'information maritime et de systèmes de combat (MICS) et d'autres sections pour les aider à mettre en œuvre leurs programmes de recherche. La première expérience du genre, connue aussi sous le nom de première expérience de combat virtuel du Canada-1 (VBE CA-1), porte sur l'incidence du partage, au sein de la coalition, des données sur les segments de trajectoire provenant de sonars passifs large bande.

L'infrastructure de la VBE CA-1, qui utilise l'architecture de haut niveau (HLA), consiste en un certain nombre d'éléments individuels de simulation (fédérés), dont certains ont été fournis par la Defence Science and Technology Organisation (DSTO) de l'Australie et d'autres ont été développés par RDDC Atlantique afin de satisfaire aux besoins de cette expérience.

L'expérience VBE CA-1 visait à déterminer quels étaient les avantages sur le plan de la connaissance de la situation du partage, au sein de la coalition, des données non traitées sur les segments de trajectoire provenant de sonars passifs large bande. On a demandé aux opérateurs de sonar de surveiller le trafic des navires dans un détroit et d'identifier les trajectoires sonar qui provenaient de cibles communes. Dans certains cas, les données sur les trajectoires sonar fournies aux opérateurs provenaient de leur propre navire et d'un second navire (de la coalition); dans d'autres cas, les opérateurs n'avaient accès qu'aux données de trajectoires sonar provenant de leur propre sonar. L'expérience a été répétée plusieurs fois, avec divers sujets, afin d'obtenir des résultats pertinents du point de vue statistique.

Le présent document traite principalement des activités techniques qui ont été complétées pour la préparation de cette expérience. Il porte sur l'élaboration de scénarios ainsi que sur la sélection, le développement et l'intégration d'éléments de simulation et sur l'essai de l'architecture.

Résultats

L'infrastructure fournie par la DSTO a été intégrée aux fédérés élaborés par RDDC Atlantique. Les processus expérimentaux nécessaires à la planification et à la réalisation d'une expérience virtuelle d'intervention humaine ont été mis à l'essai et précisés.

Dans l'ensemble, la capacité de simulation d'un environnement maritime, d'immerger des opérateurs dans l'environnement virtuel et d'obtenir des résultats pertinents du point de vue statistique sur des questions d'importance militaire a été démontrée avec succès.

Portée

C'était la première fois que le groupe SCV menait une expérience du genre et l'équipe de recherche a donc beaucoup appris durant ces travaux (intégration des éléments de simulation dans l'architecture, élaboration de nouveaux éléments, création de divers scénarios, etc.). La base de connaissance obtenue a permis de mieux comprendre l'effort à fournir pour la réussite d'une VBE. Cette expérience permettra de planifier et de réaliser les recherches futures de manière plus efficace.

Les expériences virtuelles possèdent du potentiel pour l'évaluation initiale des nouvelles technologies et des concepts opérationnels. Leur coût et le risque qu'elles présentent sont faibles par rapport à ceux associés à l'utilisation d'une frégate ou d'une sortie en mer par un groupe. Elles peuvent servir à détecter les anomalies ou les erreurs avant la prise d'engagements financiers ou de ressources pour la conduite d'expériences en milieu réel. Au fur et à mesure que l'environnement deviendra plus complexe, elles permettront non seulement de fournir des renseignements avant les essais en mer, mais aussi de réduire le nombre global d'essais nécessaires.

Recherches futures

Cette expérience est la première d'une série d'expériences menées dans le cadre du programme de partage de trajectoires sonar au sein de la coalition (CoaSTS), qui examinera le partage des données de la coalition à différents niveaux de développement. Lors des futures recherches de cette série, des outils d'analyse plus perfectionnés (de triangulation ou d'analyse des mouvements des cibles - TMA) seront fournis aux opérateurs.

Wentzell, T.E. and Mellema, G.R. 2004. Technical Preparation for an Initial Maritime Virtual Battle Experiment. DRDC Atlantic TM 2004-014 Defence R&D Canada – Atlantic.

Table of contents

Abstract i

Executive Summary iii

Table of contents vii

List of figures x

1. Introduction 1

2. Background 2
   - 2.1 High Level Architecture (HLA) 2
   - 2.2 Virtual Maritime System Architecture (VMSA) 2

3. The VBE CA-1 Team 3

4. VBE CA-1 Overview 4
   - 4.1 Motivation 4
   - 4.2 Experimental Objectives 4
   - 4.3 General Description 5
   - 4.4 Experimental Infrastructure 6
     - 4.4.1 Gameboard 7
     - 4.4.2 Motion 7
     - 4.4.3 Horizon 8
     - 4.4.4 Virtual Maritime System Execution Manager (VMSEM) 10
     - 4.4.5 Sonar (Version 1.0) 10
     - 4.4.6 Operator Probe 10
   - 4.5 Scenarios 11

5. Application Development & Customizations 12
   - 5.1 HLA Development Tools 12
   - 5.2 Sonar Federate 12
   - 5.3 Horizon Customizations 13
   - 5.4 Association Reasons Application 14

6. File Management 17

7. Federation Set-Up 19
7.1 Adding a Federate to a Federation 19
7.1.1 Example of Adding a Federate to a Federation 20

8. Scenario Development 22
8.1 Using the Scenario Generator to Develop Scenario Control Files 22
8.1.1 Using a Scenario Generator Control File in a VMSA Federation 23
8.2 Customizing the Scenario Components File 24
8.2.1 Scripting Ownship Motion 24
8.2.2 Adding Sonar Components to Platforms 25
8.3 Using the Bearing Generator to Preview Track Layout 26
8.4 Assuring Appropriate Association Opportunities 27

9. Executing the Simulation 29
9.1 Clock Synchronization 29
9.2 Potential Federation Interruptions 29
9.3 Running the Federation 30

10. Testing the Simulation 31

11. Concluding Remarks 33

12. References 35

Annex A: Log File Formats 37

Annex B: VBE CA-1 Version Fact Sheet 39

Annex C: Sonar Configuration File 41

Annex D: Horizon Initialization Files 43

Annex E: Gameboard Scenario Control File for Training Scenario 57

Annex F: VBE CA-1 Scenario Components File 61

List of symbols/abbreviations/acronyms/initialisms 63

Distribution List..... 65

List of figures

Figure 1. VBE CA-1 Experimental Run 6

Figure 2: Computer Stations for VBE CA-1 7

Figure 3. Horizon Navigational Display 9

Figure 4. Horizon Time-Bearing Display 9

Figure 6. Application for Obtaining x and y Cursor Coordinates 15

Figure 7. Association / Disasso[illegible]tion Reasoning Form 16

Figure 8. Scenario Built within Scenario Generator 24

Figure 9. Bearing Generator Displaying Tracks Relative to Ownship 27

Figure 10. Example of Track Association Opportunities 28

Figure 11. JSS Clock Synchronization Program 29

List of tables

Table 1. Horizon Initialization Files for VBE CA-1 13

Table 2. Storage Locations of VBE CA-1 Files 18

Table 3. Horizon Log File Format 37

Table 4. Sonar Log File Format 38

Table 5. VBE CA-1 Version Fact Sheet 39

1. Introduction

The Virtual Combat Systems (VCS) Group at Defence Research and Development Canada – Atlantic (DRDC Atlantic) is aiming to provide a combat system experimentation facility to assist with the research programs of scientists within the Maritime Information and Combat Systems (MICS) section and beyond. The first such experiment, termed Virtual Battle Experiment Canada-1 (VBE CA-1) to distinguish it from a series of international VBEs being performed within the Maritime Systems Group Technical Panel-1 of The Technical Cooperation Program (TTCP MAR TP-1), was completed in October 2003 and focused on coalition sharing of broadband passive sonar track data.

This experiment was the first of its kind executed by the VCS group. As such, the experimental team learned a significant amount in completing each phase of the experiment: planning, implementation, and analysis. This document will focus on the implementation phase of the experiment, particularly items of a technical nature. It is part of a collection of documents that will cover the various details of this experiment. Specifically, there are two other DRDC Atlantic documents that, along with this one, describe the various aspects of VBE CA-1:

a. Experimental Plan for VBE CA-1 [1], which covers the motivation for the experiment, and provides detailed information on the objectives of the experiment, infrastructure requirements, scenarios, and human experimentation issues; and

b. VBE CA-1: An Investigation Into Low-Level Coalition Passive Sonar [2], which examines the results and scientific value of the experiment.

As each paper is meant to stand on its own while providing a focus on a particular area, there will inevitably be some degree of overlap in each. This paper attempts to capture the technical work that took place prior to execution of VBE CA-1; it first provides an overview of the experiment and then discusses federate development, configuration and integration, scenario development, and simulation testing.

2. Background

2.1 High Level Architecture (HLA)

The High Level Architecture (HLA) [3] is a framework developed by the Defence Modeling and Simulation Office (DMSO) in the United States to promote simulation reusability and interoperability. It uses the terminology 'federate' to indicate the simulation of an individual component (a task force, a single vessel, a sonar, etc., depending on the level of fidelity required) and 'federation' to refer to a group of federates that share information with each other as part of a larger simulation. The information is shared and managed via the Run Time Infrastructure (RTI). A federate that produces information for the rest of the federation is said to 'publish' information, and a federate that uses this information is said to 'subscribe' to it. The Federation Object Model (FOM) defines what information can be published and subscribed to at run time to accomplish the objectives of the federation. A Simulation Object Model (SOM) defines the information that an individual federate can provide (publish) to a federation and the information it will use (subscribe to).

2.2 Virtual Maritime System Architecture (VMSA)

The Virtual Maritime System Architecture (VMSA) [4] is an HLA-implemented simulation infrastructure of a maritime environment. It was developed by the Defence Science and Technology Organization (DSTO) in Australia to support work on Anti-Air Warfare (AAW) ships and submarine combat systems and has been provided to Canada under a collaborative agreement through TTCP MAR TP-1.

Typical federates in this architecture will model sensor behaviours, weapons, navigation, and command and control systems. In keeping with HLA's goal of reusability, a collection of federates have been developed which may be selected from when planning an experiment and designing a federation. When new federates are required to meet the needs of an experiment, or greater fidelity is required in an existing federate, new models are created, or existing federates are modified if source code is available. VMSA uses the time management features of HLA, to ensure that simulations are repeatable in order to generate statistically relevant results.

Experiments using the VMSA architecture can support various types of investigations including evaluation of new concepts, procedures, or equipment. VMSA has been the underlying architecture for a series of human-in-the-loop experiments already undertaken: three by scientists at DSTO and two by the international TTCP MAR TP-1 group, each with a focus towards information exchange and picture compilation. Several of the TTCP nations are working towards their own implementations of VMSA to benefit their national research programs in addition to contributing to the international group.

3. The VBE CA-1 Team

The Virtual Combat Systems group at DRDC Atlantic was stood up in the summer of 2002 and for the duration of this experiment, consisted of two defence scientists, a computer scientist and a Turkish naval exchange officer. One of the aims of the VCS group is to provide a combat systems experimentation facility to assist with the research programs of groups within the MICS section and beyond. For this first experiment, the Underwater Warfare Data Fusion (UDF) group was approached for a research topic that could reasonably be tested within the group's simulation environment. It was expected that some additional federates would be required, but with a short time line and an awareness of the learning curve the group was up against, it was desirable to begin with a simple experiment that would have scientific merit of its own, but even more so as part of a more elaborate experimental programme. A defence scientist from the UDF group formulated the focus of this experiment and collaborated with the VCS group in the planning and execution of VBE CA-1. VBE CA-1 investigated the sharing of broadband passive sonar track segment data at the most primitive (unprocessed) level. It is part of a larger experimental program called the Coalition Sonar Track Sharing (CoaSTS) program [5], which will investigate coalition sharing of broadband passive sonar data at various levels of development. Following initial planning phases and identification of federation requirements, a contractor was also hired to assist with federate development.

4. VBE CA-1 Overview

4.1 Motivation

Typically, sharing of sonar data is held off until the individual vessels have been able to obtain position estimates of tracked vessels. While exchange of information at this level may prevent unnecessary depletion of available bandwidth and potential information overload of the operators, it also means that the unprocessed track data is held back even though it has potential to provide insight to coalition members in a more timely fashion. There is inevitably a trade-off between bandwidth use and operator overloading, and the timely delivery of various levels of shared data.

The CoaSTS program is a series of controlled, repeatable, human-in-the-loop experiments to investigate the influence of shared broadband passive sonar track data at various levels of development. This experiment, VBE CA-1, was the first in this series and investigated the sharing of broadband passive sonar track data prior to any operator or automated processing. The objective was to establish whether or not there was any benefit to the operational picture as a result of this lowest level of track sharing. For this first experiment, the operator was not provided with any tools for estimating positional information or fusing tracks from sonar on different ships.

4.2 Experimental Objectives

There were two primary objectives for this experiment. The first was to test the hypothesis that the sharing of sonar track data with a coalition partner is beneficial even without the availability of Target Motion Analysis (TMA) tools, triangulation tools, or any other method of electronically combining the track information from the two partners. That is, the coalition partner's track data could be beneficial as an independent display, allowing the operator only visual processing and comparison of the information available.

The second goal of the experiment was a discovery exercise to gain insight into the rationale used by a sonar operator when deciding to associate or disassociate tracks. This understanding is expected to be useful for future development of programs that provide an operator with track association recommendations, thus easing the demand on the operator and/or increasing the operator's confidence in his decisions.

An additional goal of VBE CA-1 was to successfully put together the architecture for this experiment and demonstrate the successful implementation and completion of an experiment within this environment with an end product of statistically relevant results.

4.3 General Description

For this initial experiment, military and civilian members of DRDC Atlantic were recruited to act as sonar tracking operators for a frigate, tasked to monitor the traffic in a narrow strait. There was no requirement for the operator to navigate the ship since all vessels, including the ownship, were on predefined paths within the strait to ensure repeatability of the experiment.

A navigational chart was provided to the operators that indicated the location of their ownship as well as that of an allied vessel within the strait. A time-bearing plot of ownship sonar tracks was considered to be the 'operational picture' for this experiment. Since sonar tracks break up over time due to background noise, other traffic in the strait, etc., the specific assignment of the operator was to observe the (broken) sonar tracks and determine which track segments belonged to a single target vessel. If two or more tracks were believed to have originated from the same vessel, the operator was asked to associate the segments. If the operator were to later decide that an association previously made was incorrect, the tracks could be disassociated back into their constituent segments. Since the experimental team was interested not only in the operator's performance in terms of making these associations, but also in understanding why these associations were made, a question box would pop up on the screen each time such a decision was made and ask the operator to indicate the reason for the decision.

Each operator was asked to participate in two types of scenarios. In the solo scenario type, the operator would have only the navigational chart and the ownship sonar track data as described above. In the coalition scenario type, the operator would also have sonar track data from the allied vessel for visual reference and comparison with the ownship tracks. However, no TMA tools or triangulation tools were provided.

During a typical experimental session, two operators (one solo, one coalition) worked independently through a single scenario. They would then switch positions for a second scenario run. By having some operators begin with the solo session and others begin with the coalition session, potential biases brought on by learning were avoided. The scenario file controlling the location of the vessels in the strait was different for both runs as well. Data collected throughout each experiment included:

a. mappings of which sonar tracks belonged to which vessels,
b. the time and details of tracks which were associated and disassociated, and
c. the reasons indicated for making decisions.

The formats for these files are included in Annex A.

A snap shot from an actual run of VBE CA-1 is shown in Figure 1. The solo subject is in the front of the picture with two displays and the coalition subject is in the rear with three displays.

Figure 1. *VBE CA-1 Experimental Run*

Following the experiment, a series of metrics were used to analyse the data and determine whether or not there was any statistically relevant difference in the performance of the solo and coalition operators. A complete description of this experiment can be found in [1] and the results are reported in [2].

4.4 Experimental Infrastructure

The experimental infrastructure used for VBE CA-1 was based on the VMSA architecture. The simulation ran on four computers, each equipped with dual 2.0 GHz Xeon processors and 1 GB of RAM, on a 100 Mb/s network using a private switch.

A single run of VBE CA-1 was designed to support testing with two subjects. A Solo Operator was presented with only two displays: ownship sonar tracks in time-bearing format and a plan view of the strait showing the position of ownship and the coalition vessel. A Coalition Operator had an additional display of allied vessel sonar tracks in time-bearing format.

A Run Director was also required for each experiment to monitor the progress of the simulation and answer questions posed by the subjects. The subject stations were set back-to-back so that neither subject was able to see what the other was doing.

The set-up for VBE CA-1 is shown in Figure 2. The dotted boxes indicate the simulation components contained on a single computer.

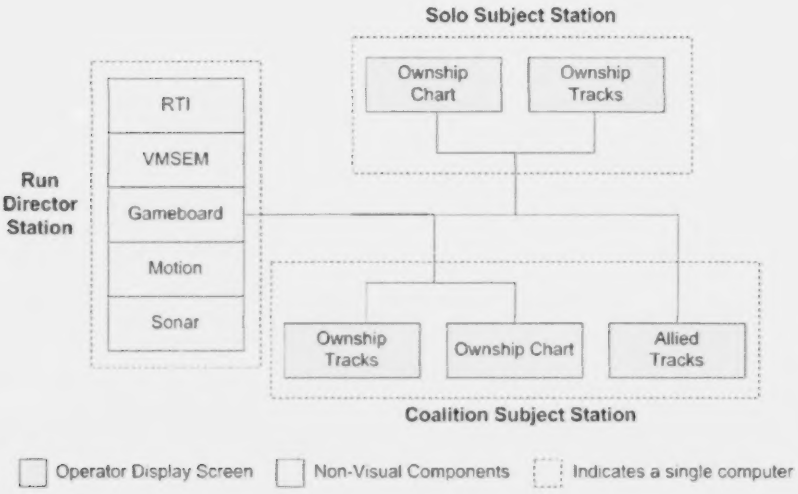

Figure 2: *Computer Stations for VBE CA-1*

Some existing VMSA components were reused in this experiment. These included Gameboard, the motion federate, Horizon, and the Virtual Maritime System Execution Manager (VMSEM). In addition to these, a new sonar federate was developed and integrated. A description of these components follows and Annex B shows the version information pertinent to this experiment and required for its reproduction.

4.4.1 Gameboard

Gameboard instantiates all vessels in the VBE CA-1 scenarios; through the processing of an XML scenario control file it publishes information about where each vessel should be positioned within the strait at various times throughout the scenario, and the course and speed it will have at that time. The XML file is produced prior to execution using the Scenario Generator tool. The Scenario Generator is, as the name indicates, a tool for generating scenarios, and will be discussed in the scenario development section of this paper.

4.4.2 Motion

The motion federate subscribes to the information from Gameboard and uses it to determine the position, course, and speed of each vessel at all times throughout the scenario (Gameboard only provides this information for specific points in time). The motion federate manoeuvres the vessels along the determined path.

4.4.3 Horizon

Horizon [6] is a track-hosting and management system used in VMSA. It was developed by Innovation Science Pty Ltd for DSTO with a focus on submarine applications. While some of the controls and options available within Horizon are not applicable to this experiment, the ability to customize the interface made it a good choice for VBE CA-1.

Horizon has two main display screens and a series of plug-ins that provide capabilities to manage sensor tracks and track segments. The navigational display shows the operator where the ownship and allied vessel are positioned within the strait. Control buttons allow the operator to zoom in on objects or locations and to hide fused (associated) or raw track segments. The cursor position plug-in can be used to determine the latitude, longitude, bearing and range of the cursor position within the navigational display. The navigational display used in VBE CA-1 is shown in Figure 3.

The time-bearing display provides a history of all tracks received in the last 30, 60, or 90 minutes. It is within this display that the operators were expected to concentrate their efforts and make the decisions to associate (or disassociate) tracks. The plug-ins available to the operator in this experiment were a fusion plug-in to associate and disassociate tracks, annotation generation and maintenance plug-ins for creating and managing notes at various locations in the display, and the cursor position plug-in that could be used to determine the time and bearing of the cursor within the time-bearing display. A single copy of Horizon provides access to both the navigational display and the time-bearing display, but in the version used for the experiment, they could only be viewed one at a time. In order to allow the operator to view both displays side-by-side, individual copies of Horizon were used. In total, the experiment used five copies of Horizon: two for the solo operator station and three for the coalition operator station. The time-bearing display used in VBE CA-1 is shown in Figure 4. Newer versions of Horizon allow a navigational display and time-bearing plot to be displayed simultaneously; so future experiments will require fewer copies of Horizon per required display screen.

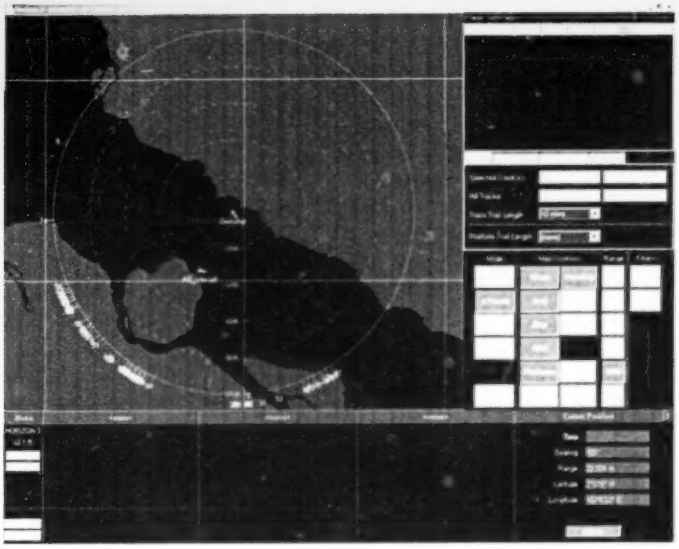

Figure 3. *Horizon Navigational Display*

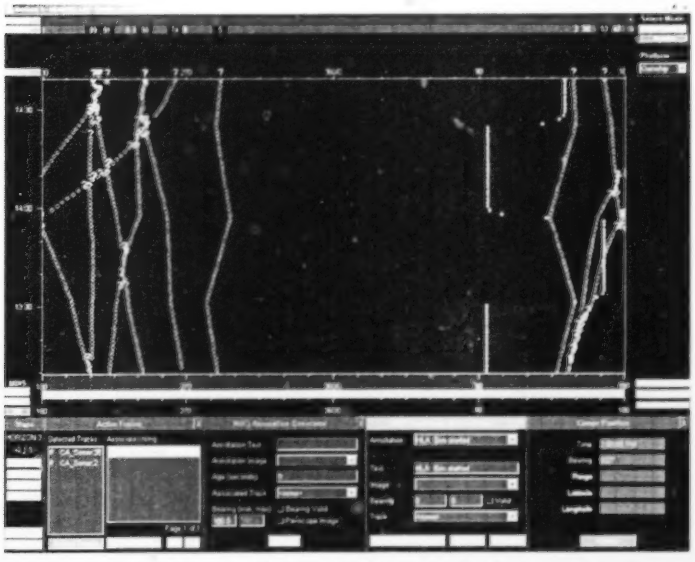

Figure 4. *Horizon Time-Bearing Display*

4.4.4 Virtual Maritime System Execution Manager (VMSEM)

The VMSEM controls the overall execution of the federation. It ensures that all federates have joined the federation before the simulation is allowed to proceed. It also provides the capability to run a scenario multiple times within a single federation execution, although this was not utilized within VBE CA-1 since the scenarios differed for each session, and the subjects required a break before moving from one 2-hour session to the next. For a detailed look at execution management within VMSA, refer to [7].

4.4.5 Sonar (Version 1.0)

Not all components required for this experiment were readily available within the VMSA architecture. A very critical component of this experiment was the sonar federate and while VMSA does include a sonar federate, it did not provide the realism that was demanded by this experiment. Thus, a new sonar (tracker) federate was developed. This federate subscribes to the position and type of all vessels (including the ownship) to determine what tracks to produce. The tracks produced break up over time and appear representative of those that are expected in a real-world situation. It was not necessary for the sonar to work through all the steps of a real-world sonar – only that the output tracks were realistic for the given scenario. The development of this federate is discussed further in Section 5.2.

4.4.6 Operator Probe

A mechanism was required to capture the operator decision criteria as association and disassociation choices were made. Neither verbal questioning nor a Dictaphone were reasonable solutions since two operators would typically be participating in each run and this verbalization would not only be distracting to the operators but could also influence their choices. A pen-and-paper approach was also rejected since it would have been nearly impossible to know which association was being referred to without an accurate way to capture the association time.

The chosen solution was to provide an option box that would pop up on the screen each time an association or disassociation was made and electronically capture the time of the decision and log the operator's answer directly to a text file. This application will be discussed further in Section 5.4.

4.5 Scenarios

Five scenarios were developed for VBE CA-1. One scenario was designated as the training scenario and was used to demonstrate the use of Horizon to the volunteer subjects. Subjects were allowed to test their understanding of their tasks following the training session, also using the training scenario. Two other scenarios were required since each operator had to complete two runs (one solo, one coalition). To be prepared for any unforeseen interruption to a session, two back-up scenarios were developed and kept in reserve so that it would not be necessary to restart the simulation with the same scenario.

The scenarios developed for VBE CA-1 were set in the Strait of Malacca (as was shown in Figure 3). The ownship and allied vessel were tasked to monitor the vessel traffic in the strait and were already in the midst of the strait when the scenarios began. Both vessels were pre-programmed to make turns approximately every 15 minutes. Traffic in the strait consisted of eight merchant vessels, and all were set to travel at a predefined speed and course throughout the scenario.

5. Application Development & Customizations

This section discusses the creation and customization of the experimental components. It also explains how to start the federation for demonstrations or further runs and what potential interruptions to be aware of.

5.1 HLA Development Tools

The federate development required for VBE CA-1 brought about investigations into some HLA development tools currently on the market. The two tools looked at in greatest detail were Simplicity and S2Focus. GreenLight Software Solutions conducted the investigation and presented the results to the VCS Group [8]. The overall conclusion was that Simplicity was valuable as a planning tool (particularly for building a federation from scratch), but resulted in undue complication in terms of developing a single federate to add to a federation and an existing FOM. As for S2Focus, its strength was deemed to be as a visualization tool, rather than an HLA development tool.

5.2 Sonar Federate

The sonar federate that existed within VMSA was not sufficiently complex for VBE CA-1, so a higher-fidelity broadband passive sonar tracker federate was developed by GreenLight Software Solutions. It was not necessary to model the explicit details of a broadband passive sonar. Rather, the requirement was to produce realistic-looking bearing-over-time track segments (i.e., tracks that fade in and out over time based on surrounding acoustic influences and the proximity of targets to the sonar-hosting vessel) that could be presented visually to the operator and accepted as representative of the real world.

The sonar federate subscribes to the position and type of each vessel in the simulation. The type of vessel corresponds to signal strength that is defined in the sonar configuration file (Annex C); the position of the vessel is required to calculate the transmission loss of the signal (using spherical-cylindrical spreading). The sonar federate uses the position of all vessels to determine interference of a particular signal, and also considers the effects of randomized ambient noise.

When the signal excess of the received signal from a target exceeds a specified threshold at least m times within n simulation time steps, a sonar track segment is initiated at a bearing corresponding to a direct path arrival from that vessel. Once initiated, the track is regularly updated so long as the signal excess does not fall below the specified threshold for more than a specified number of consecutive time steps.

The details of each track are maintained in a log file, which describes when tracks are initiated and lost, bearings over time, target identity, etc.. The format for the sonar federate log file is shown in Annex A.

5.3 Horizon Customizations

Horizon, the track hosting and management system used for VBE CA-1, is packaged with many features that were not required for this experiment. Horizon was configured to hide such features where possible to minimize operator confusion and prevent the use of tools not permitted in this initial experiment.

Customization of Horizon was achieved by editing the initialization files. These files have extensions *.ini* and can be found in the initialization file directory, *Horizonr2.1.5\ini*. They are text files and can be opened with any text editor such as Notepad. Table 1 gives a brief description of the purpose of each file and how each file was customized (i.e., changed from the default version obtained with Horizon) for this experiment. The complete Horizon initialization files relevant to VBE CA-1 can be found in Annex D.

Table 1. *Horizon Initialization Files for VBE CA-1*

INITIALIZATION FILE	GENERAL PURPOSE	CUSTOMIZATIONS
Filters.ini	Defines the graphical symbols that are used to distinguish tracks of one sensor type from another and provides default filters that appear in the application upon start-up. Filters allow the user to view a certain selection of tracks at a time, based on sensor type, hostility, etc.	Raw sonar tracks were assigned a circular symbol and fused tracks were assigned a lightning bolt symbol. Filters were designed for displaying only fused tracks or only sonar tracks for the selected vessel.
Fusion.ini	Defines active fusion algorithms that can be chosen when associating tracks.	Only the 'Combine All Tracks' algorithm was offered as it provided the most obvious resultant track and choosing an appropriate fusion algorithm was not relevant to the experiment.
Hla.ini	Defines how Horizon will interact with the HLA federation.	The following were defined: the name of the federation, the name of the Horizon instance, the location of the SOM and FOM files, and which platforms should appear in the Horizon plan view display as allies.
Map.ini	Defines parameters for the map display.	Defined the default latitude and longitude where the map display centred itself upon start up (in the Strait of Malacca) and the values for the range selection buttons.
Plugin.ini	Defines which plug-ins are used, whether they are displayable (graphical) or not, the locations in which to display the plug-ins, and plug-in defaults and parameters.	Edited to include only the plug-ins relevant to VBE CA-1. Defined the Active Fusion, Annotation Generator, Annotation Maintenance, and Cursor Position plugins to be the defaults available at start-up. Text and binary loggers were turned on so that data logs would be created. Default locations for data files were also set.
Recorder.ini	Defines what the binary logger records.	Binary recorder was set to record all data possible.
Roles.ini	Defines groups of four plugins that can be displayed concurrently in Horizon.	No customization was required since only four plugins were used in the experiment and these are defined in the plugin.ini file.
Trackfusion.ini	Defines application wide parameters, including units of measure.	No customizations were required.
Vberecorder.ini	Defines parameters for the text logger.	Recording period was set to 15 seconds. Unique file names for each file were requested so that overwriting would not occur with successive runs of the experiment.

5.4 Association Reasons Application

Capturing the thought process of operators each time a choice is made to associate two or more tracks, or to disassociate an association previously made, is no small task. In VBE CA-1, each operator had the opportunity to make 100 or more associations in a 2-hour session, so it was not realistic to think operators could recall their motivation for each choice after the experiment. Therefore capturing their reasoning needed to be more immediate. Since two operators were running the experiment at the same time with the same scenario, questioning an operator verbally could have cued the other about what associations to make or break. This meant that there needed to be an electronic question form that the operators could use to record their reasoning. They were given the option of selecting a response from a list of pre-set potential answers, or entering a reason that was not included in the list.

In Horizon, all operator association decisions are completed with a mouse-click on either the green check mark button (to associate) or the red 'x' button (to disassociate) of the Active Fusion plug-in, both shown in Figure 5. So each time one of these buttons was clicked, it was necessary to have the question form appear. The source code for the Active Fusion plug-in was not available at the time of the experiment, and therefore these buttons could be recognized as nothing more than graphics on the screen (i.e., the buttons are only known as buttons within the Horizon framework, not to Windows).

To get around this, a somewhat 'sneaky' approach had to be taken. First, a simple Visual Basic (VB) application was created that would retrieve the x and y coordinates of the cursor on the screen as shown in Figure 6. With Horizon open and maximized (as it was for the experiment), the upper left and lower right coordinates of both buttons (the check mark and the 'x') were determined and recorded.

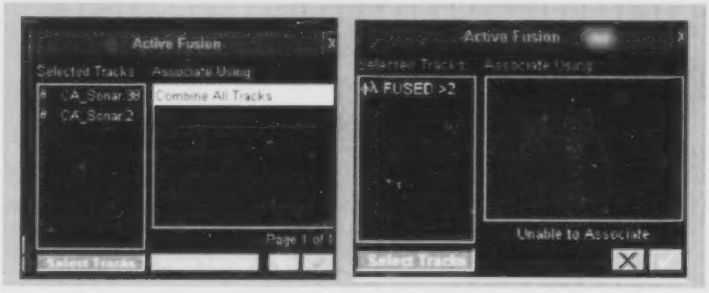

Figure 5. *Active Fusion Plug-in (Association / Disassociation)*

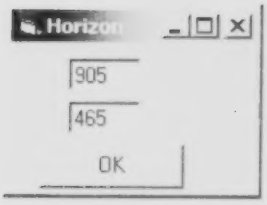

Figure 6. *Application for Obtaining x and y Cursor Coordinates*

An entry form was then created as shown in Figure 7, also using VB. The pre-set option list is configurable through an initialization file. The text box was provided to enter a reason that was not already available. This way, operator response was not limited. When the operator clicked the OK or Cancel button, the current selection was written into a text file, including any text in the text box, and the form was re-hidden.

Using the coordinates of the Horizon buttons obtained through the cursor position application, this form was set to capture all mouse click events and check to see if the cursor position at the time of the click was within the coordinate range of either of the two buttons. If it was, the form was brought up on top of the Horizon application to accept input from the user.

When running (or testing) a federation on multiple computers, it was essential that each computer running this 'Enter Reason' application was set to use the same resolution so that the coordinate ranges for the buttons were consistent. Alternatively, the application could have been configured separately for each computer, but this approach would have required configuration control.

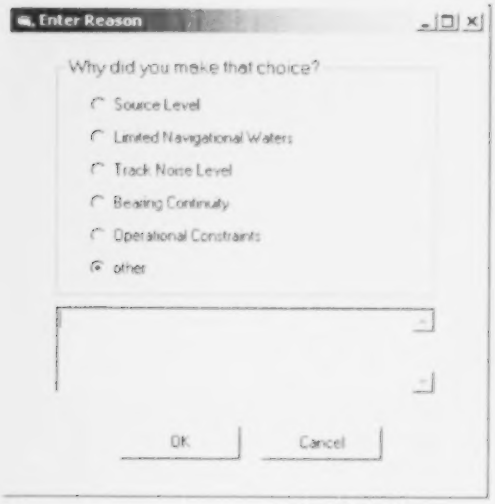

Figure 7. *Association / Disassociation Reasoning Form*

Following the experiment, a support contract was let to Innovation Science to assist DRDC Atlantic with enhancements to the Horizon interface. Under this contract, embedding such a form as part of the Active Fusion plug-in is now a possibility. For the experiment, however, the above solution was acceptable.

6. File Management

There were many files that had to be dealt with when running a VBE CA-1 simulation: configuration files, start-up scripts, federate files, scenario files and output files. A shared drive, drive v:, was created on the Run Director's computer and all other computers were supplied with a mapping to this drive. This way, there was a common repository of files accessible from each workstation. All configuration files, start-up scripts, and federate files were stored on this drive. Shortcuts to the appropriate start-up scripts were placed on the desktops of each workstation. The sonar log file was stored directly on the v: drive when a simulation was run. However, the Horizon output files and the Enter Reason output files were stored on the local hard drives, c:. There were a couple of reasons for this. First, each simulation resulted in six Horizon logs (three binary and three textual) and three Enter Reason logs. Placing them all on v: could have resulted in confusion over which file originated from which source. Second, and most importantly, it reduced the demand on network resources. Real-time performance was a significant issue for VBE CA-1 and so minimizing network traffic was important. After the completion of each experimental run, the local log files were renamed to indicate which workstation they originated from and were copied into a folder on the v: drive. The sonar configuration and log files and the scenario control file were also placed in this folder so that all information relative to the run would be available from one location. Table 2 shows the default locations of the files most significant to VBE CA-1. Text in italics indicates that there are multiple folders or files of this format, and is a placeholder for the actual file or federate name, date or time.

Table 2. *Storage Locations of VBE CA-1 Files*

FILE TYPE	FILE(S)	LOCATION
Configuration Files	Horizon initialization files: *FileName*.ini	V:\VBE_CA-1*HorizonFolderName*\ini\
	EnterReason configuration file: reasonsconfig.txt	V:\VBE_CA-1\Reasons\
	Sonar configuration file: sonar.cfg	V:\VBE_CA-1\VMSA\federates\sonar\bin\
Start-Up Scripts	Federation start-up files: *FileName*.bat	V:\VBE_CA-1\bin\ (with shortcuts from workstation desktops)
Federate Files	Federate SOM file: *FederateName*.omt	V:\VBE_CA-1\vmsa\federates*FederateName*\bin\
	Compiled federate code: *FederateName*.bat	V:\VBE_CA-1\vmsa\federates*FederateName*\bin\
	Library files	V:\VBE_CA-1\vmsa\federates*FederateName*\lib\
Scenario Files	VMSA scenario components file: VBECA1.xml	V:\VBE_CA-1\VMSA\scenarios\
	Gameboard scenario control files: *ScenarioName*.xml	V:\VBE_CA-1\VMSA\federates\Gameboard\scenarios\drdca\vmsa\gc\xml\
Output Files	Horizon binary log files: horizon_data_*Date_Time*.dat	C:\binary_log_files\
	Horizon text log files: horizon_data_*Date_Time*.txt	C:\text_log_files\
	EnterReasons log files: *Date_Time*.txt	C:\reasons\
	Sonar log file: sonar-*Date_Time*.txt	V:\VBE CA-1\federates\sonar\bin\

7. Federation Set-Up

Once the federates have been chosen, created, and/or modified, a federation needs to be set up that includes the necessary federates. The scenario components file indicates which federates the Execution Manager will expect to have participating in the federation. Batch (.bat) files are used to start the RTI, and also to start each federate.

7.1 Adding a Federate to a Federation

Adding a new federate to a federation is a common requirement when setting up a new simulation. The following describes how a new VMSA 2.0-compliant federate could be added to the VBE CA-1 federation. While some details may be federate-dependent, the main considerations are captured here.

a. To be consistent with the other federates, store the federate folder in the directory V:\VBE_CA-1\VMSA\federates.

b. In the V:\VBE_CA-1\VMSA\scenarios\ folder, open the scenario components file, VBECA1.xml, in edit mode (right-click on the file and select the "edit" option), and add:

    <ParticipatingFederate Name="*NewFederate*" />,

    where *NewFederate* is the name of the federate being added. At run-time, this tells the Execution Manager not to proceed with the simulation until *NewFederate* has joined the federation.

c. In the VBE_CA-1\vmsa\bin folder, create a batch file to execute this new federate and call it *Start_NewFederate*.bat. While it is difficult to generalize the exact format of this file, the idea is to open the bin directory of the federate that is being added (where the application executable should be stored), and call the application. It should look something like this:

    cd /d %FEDS%*NewFederate*\bin
    call "*NewFederate*.bat"

    @ECHO OFF
    IF errorlevel 1 pause

    where FEDS is a previously defined (in the setenv.bat file in the VBE_CA-1\vmsa\bin folder) environment variable, which identifies the location of the federate folders (V:\VBE_CA-1\VMSA\federates in the case of this experiment). There may also be some additional federate-specific parameters required for this batch file.

There is a batch file of this type for each federate. Clearly it would be cumbersome to execute a file for each federate for each run (particularly as the federation becomes larger). Thus, there are higher-level batch files (one for each computer that will host one or more federates) that call each of the individual batch files for federates that it will host. This leads to the final step of adding a federate.

d. Open the batch file in VBE_CA-1\bin\ that corresponds to the computer that will host the new federate. Add the following line:

start /MIN "*WindowName*" "Start = *Start_NewFederate*.bat"

where *WindowName* can be any string of characters chosen to appear at the top of the federate window when it executes.

7.1.1 Example of Adding a Federate to a Federation

To provide a more detailed example, the case of adding a particular federate to the VBE CA-1 federation will be used. Another VMSA 2.0 federate provided to the VCS Group by DSTO is called the Virtual Maritime System Situational Display (VMSSD). This federate provides a god's-eye view of the simulation, showing the kinematics (most notably, position) of all platforms in the scenario at any time.

a. Place the VMSSD federate folder in the directory V:\VBE_CA-1\VMSA\federates.

b. In the V:\VBE_CA-1\VMSA\scenarios\ folder, open the scenario components file, VBECA1.xml and add:

```
<ParticipatingFederate Name="VMSSD" />,
```

c. In the VBE_CA-1\vmsa\bin folder, create a batch file called Start_VMSSD.bat and type the following:

```
cd /d %FEDS%\VMSSD\bin
call vmssd.bat -fx %FEDERATION_NAME% -fn VMSSD -fed
%FOM% -som %SOM_DIR%\VMSSDSOM.omt

@ECHO OFF
IF errorlevel 1 pause
```

where FOM and SOM_DIR are environment variables representing the location and file name of the VMSA FOM and the directory containing the VMSSD SOM, respectively.

d. Open the batch file in VBE_CA-1\bin\ that corresponds to the Run Director's computer, so simulation truth can be monitored by the experimenter. Add the following line:

start /MIN "Situational Display" "Start = Start_VMSSD.bat".

8. Scenario Development

For each scenario, there are two types of files that determine the way the scenario plays out. The scenario components file identifies which platform is to be recognized as the ownship in each Horizon instance and what sensors are on each platform. The scenario control file provides information to the simulation which instantiates the vessels in the strait and specifies all of the turns and speed changes that occur throughout the scenario. For all five scenarios in VBE CA-1, the scenario components file was the same, however, the scenario control file was unique to each scenario. This section discusses the creation of these files, as well as the tools and methods used to create scenarios appropriate for VBE CA-1.

8.1 Using the Scenario Generator to Develop Scenario Control Files

The Scenario Generator [9] tool was created to assist with the development of scenario control files for use with the VMSA architecture. While an XML file to instantiate and control the vessels could theoretically have been created using a simple text editor, it would have been very difficult to identify the desired positions, courses and speeds without some sort of visualization tool. Scenario Generator provides a graphical user interface that allows the user to develop a scenario through a series of input boxes (to name vessels, define the type of vessel, set speeds, etc.) and mouse clicks (to identify positions and course changes). It generates an XML scenario control file, which can be read and interpreted by the Gameboard federate and translated into information comprehensible by a VMSA federation. The scenario control file used for training in VBE CA-1 is found in Annex E.

Using the Scenario Generator is quite straightforward since a series of screens, buttons, and help tips guide the user through the scenario development. Although it would have been possible to pre-script only a selection of the platforms and control the others through the simulation, all VBE CA-1 platforms (including ownship and the allied vessel) were pre-scripted to ensure repeatability of the experiments.

The main challenge with this stage of scenario development was figuring out how to coordinate the latitude and longitude positions in the Scenario Generator with those in Horizon. Scenario Generator is unaware of geographical locations and scenarios are simply built on graphical images. For VBE CA-1, an image of the Strait of Malacca was imported into Scenario Generator.

The center point of the image in latitude and longitude and the x and y range of the image in kilometres (or nautical miles) are required inputs for Scenario Generator. To ensure consistency between Horizon and Scenario Generator (so that what looks like water in one program is water in the other), a screen capture of the Strait of Malacca in Horizon was used as the image for the Scenario Generator.

The specific steps to create a scenario in Scenario Generator that will translate correctly into Horizon are as follows:

a. In Horizon, locate the desired geographical area (e.g., the Strait of Malacca) using the Lat/Long map control. Zoom out/in so that the entire area that could be traversed throughout the scenario is visible.

b. It is necessary to accurately determine the centre point of the map. With the Cursor Position plug-in active, place the cursor as close as possible to the centre. Record the latitude and longitude shown in the plug-in. Using the Platform Generator plug-in, create a new platform with latitude and longitude as recorded (it will be necessary to convert from degrees-minutes-seconds to a decimal value) and a speed of zero. Use the Object map control option to centre the screen on the new platform. At this point the exact centre of the screen is known to be the recorded value.

c. Make sure the Rings map control button is selected (this places range rings on the map). Perform a screen capture (Alt+Prnt Scrn) and crop the captured picture (using Microsoft Paint or the like) to a square with borders tangent to the outermost range circle of the map. At this point the range of the map (in both directions) is *twice* the value of the ring radius.

d. Save the picture in .jpg format.

e. In Scenario Generator, insert this picture as the background. Select the Maintain Aspect Ratio check box. Enter the centre point and the x range value and click the Set Gameboard button. Design the scenario.

An example of a completed scenario developed within the Scenario Generator is shown in Figure 8. This was the training scenario used for VBE CA-1.

8.1.1 Using a Scenario Generator Control File in a VMSA Federation

To use a scenario control file generated by the Scenario Generator in a VMSA federation, place the file in the Gameboard federate's XML folder (for VBE CA-1, this folder was v:\vbeca1\vmsa\federates\Gameboard\scenarios\drdca\vmsa\gc\xml). Edit the name of the scenario control file in the gameboard.bat file (found in the Gameboard federate bin directory, v:\vbeca1\vmsa\federates\Gameboard\bin) so that it is consistent with the name of the new file.

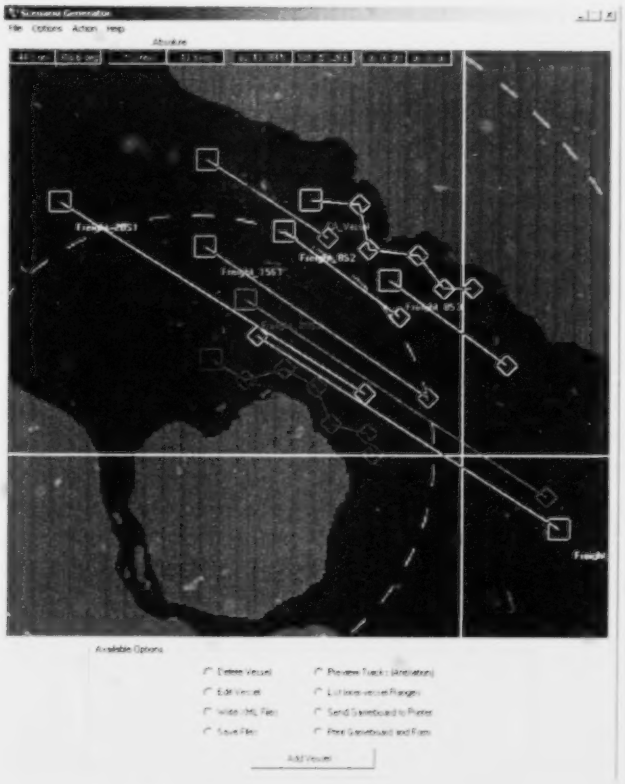

Figure 8. *Scenario Built within Scenario Generator*

8.2 Customizing the Scenario Components File

It is often the case that when there is not an interface to assist with the development of a particular file, the file may be most easily created by editing a copy of an already existing file (perhaps from a previous simulation). Annex F includes the scenario components file that was used for VBE CA-1; this file can serve as a starting point for the development of future scenario component files.

8.2.1 Scripting Ownship Motion

In past VMSA simulations, the ownship was typically manoeuvred through the use of a helm federate, which allowed the operator to change the speed and course of the ownship throughout the simulation. In the case of VBE CA-1, where repeatable runs were key to obtaining statistically valid results, the ownship motion was scripted throughout the scenario. The ownship

motion was input into the Scenario Generator in the same manner as every other vessel. Since all vessels were created the same way, Horizon could not immediately know which of the pre-scripted vessels was intended to be the ownship.

The ownship was identified to Horizon in the VBE CA-1 scenario components file. In VMSA, platforms such as a frigate are referred to as composite entities, and component entities are items that are added onto a platform, such as a sonar, radar, or command and control systems. The following script elements were required in this file to indicate that CA_Vessel (named in the Scenario Generator) was to be considered the ownship in the Horizon instance named Horizon_own:

```
1       <CompositeEntity Type="SeaSurface" Time="0.0"
        Name="CA_Vessel" FederateName="Gameboard"
        ConfigFile="Gameboard.cfg">
2       <ComponentEntity Type="CommandAndControlSystem"
        Name="CA_Horizon" FederateName="Horizon_own"
        ConfigFile="horizon.cfg">
3       <ComponentEntityAttribute AttributeName="ComponentName">
4       <AttributeEntry
        AttributeValue="CommandAndControlSystem...."
        AttributeType="s"/>
5       </ComponentEntityAttribute>
6       </ComponentEntity>
7       </CompositeEntity>
```

In line 1, the composite entity, CA_Vessel, is defined as a sea surface platform and identified as ownship. In lines 2-6 a command and control system (named CA_Horizon) is added to CA_Vessel. The federate name, Horizon_own, is the name by which VMSA will know the Horizon instance. For VBE CA-1, CA_Vessel appeared as the ownship in multiple instances of Horizon. Thus, lines 2-6 were repeated (before line 7) multiple times and the command and control system name and Horizon instance name were edited accordingly.

8.2.2 Adding Sonar Components to Platforms

Vessels cannot be fit with sonar through the Scenario Generator interface. For VBE CA-1 both the ownship and allied vessel required a passive towed array sonar. This was accomplished by adding sonar component entities to the composite entities in the scenario components file.

To add a sonar to CA_Vessel, lines 7-11 below must be added to the VBE CA-1 scenario components file:

```
1       <CompositeEntity Type="SeaSurface" Time="0.0"
        Name="CA_Vessel"
        FederateName="Gameboard" ConfigFile="Gameboard.cfg">
```

```
2       <ComponentEntity Type="CommandAndControlSystem"
        Name="CA_Horizon" FederateName="Horizon_own"
        ConfigFile="horizon.cfg">
3       <ComponentEntityAttribute AttributeName="ComponentName">
4       <AttributeEntry
        AttributeValue="CommandAndControlSystem...."
        AttributeType="s"/>
5       </ComponentEntityAttribute>
6       </ComponentEntity>
7       <ComponentEntity Type="SensorSystem.SonarSystem"
        Name="CA_Sonar" FederateName="Sonar"
        ConfigFile="sonar.cfg">
8       <ComponentEntityAttribute AttributeName="ComponentName">
9       <AttributeEntry
        AttributeValue="SensorSystem.SonarSystem..."
        AttributeType="s"/>
10      </ComponentEntityAttribute>
11      </ComponentEntity>
12      </CompositeEntity>
```

The complete file in Annex F shows sonar components added to the ownship and allied vessel.

8.3 Using the Bearing Generator to Preview Track Layout

When developing a scenario in the Scenario Generator, it is difficult to imagine what the relative bearings of all vessels are to a particular vessel (in particular, the ownship) over time. The Bearing Generator tool was developed by GreenLight Software Solutions to read the XML scenario control file generated by the Scenario Generator and instantly generate the relative bearings of all vessels relative to a chosen vessel over time. Figure 9 shows the relative bearings of all vessels in the training scenario to the ownship.

This tool had a number of applications when developing VBE CA-1 scenarios. First of all, it was the only way to get an idea of how the tracks would appear in a time-bearing display in Horizon, without running the entire simulation environment for two hours. It allowed the scenario developer to identify the number of times the tracks would cross over each other and in turn evaluate the overall complexity of the scenario. If a track did not look as hoped, the developer could go into the XML file and delete or edit the position, course or speed of the corresponding vessel. Alternatively, the developer could edit the vessel kinematics in the Scenario Generator. (It should be noted that any changes made directly to the XML file will not be reflected if the scenario is reopened in the Scenario Generator. Thus, if further editing may be desired at any point within the Scenario Generator, the developer should make all changes within this environment and then regenerate the updated files).

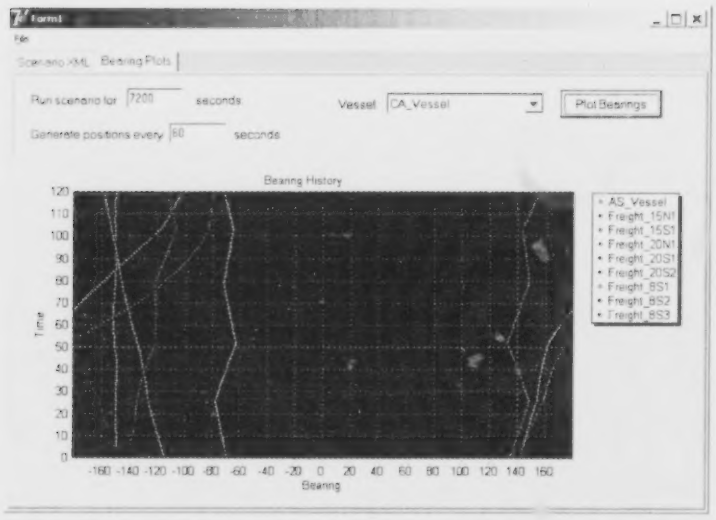

Figure 9. *Bearing Generator Displaying Tracks Relative to Ownship*

When developing a scenario it can be useful to create multiple vessels that could be used as the ownship and then decide which one is most desirable based on the track bearings that are observed in the Bearing Generator. The alternative ownships can then be deleted or left as traffic.

The only interesting element of a scenario that cannot be observed within the Bearing Generator tool is how the tracks will break up over time as a result of interfering signals, background noise, etc. These breaks are best observed by actually running the scenario through the simulation environment. Nonetheless, this tool gives the developer an understanding of what should be expected in the Horizon display and gives the opportunity to tweak things before investing the effort in simulation testing.

8.4 Assuring Appropriate Association Opportunities

In this experiment, the main task of the operators was to observe the tracks over time and decide which tracks were following a single target. At any given time, only one track could exist for a single target, so a correct association necessarily involved tracks that were distinct in time. For example, in Figure 10, it is possible that track segment T1 could be correctly associated with segment T3, but not with T2. (The yellow question marks amidst the tracks indicate the beginning of new tracks.)

Target detection is dependent upon the location and source level of the target vessel (where source level is based on the type of platform), the ambient noise level, and the degrees of randomness in the background noise. To increase or decrease the

likelihood of target detection, source level for each target type and ambient and background noise can be modified within the sonar federate configuration file (shown in Annex C). Vessel type and location throughout the scenario is scripted prior to scenario execution using the Scenario Generator.

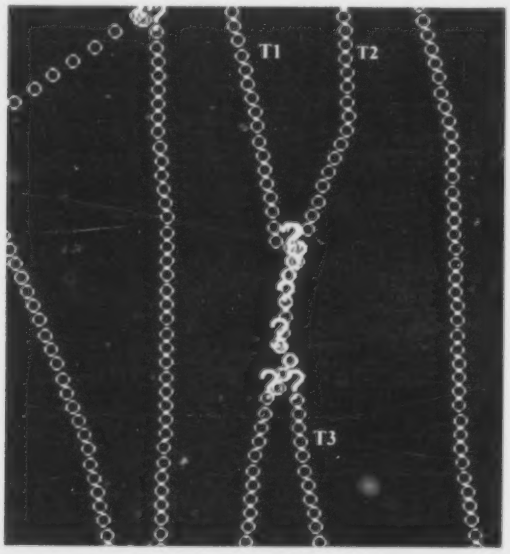

Figure 10. *Example of Track Association Opportunities*

The number of track disruptions throughout the simulation is the primary contributor to the difficulty of the scenario. Too many tracks could overwhelm the operator, while too few could result in a scenario that is uninteresting and does not produce enough data for statistical analysis. To see how tracks in a given scenario would break up over time, it was necessary to run the scenario through the simulation environment for the duration of the scenario (2 hours). Watching the simulation progress over this time would be a significantly tedious task, so a screen recording software program called Camtasia was utilized. While a scenario was run, the operator displays for both the ownship and allied vessel were recorded. After the scenario was complete, the video was played back in Camtasia at a chosen speed. The difficulty of the scenario was then judged and adjustments were made to the scenario control file (and/or the sonar configuration file, although this file had to remain constant for all scenarios) and tested again as necessary. This remained a rather subjective task, but in the end five scenarios were developed that presented the operator with approximately 150 track segments each over a 2-hour period. Given only nine vessels in a scenario (including the allied vessel), the ownship operator would then (ideally) correctly associate the 150 tracks into a refined collection of nine tracks. Of course, this perfect solution is unlikely, but it is this solution that the operator performance metrics measure the actual results against.

9. Executing the Simulation

This section discusses the method for synchronizing the clocks of all computers used in the simulation, potential interruptions to a federation execution, and how to actually make the simulation run once all the federation components, scenarios and script files are in place.

9.1 Clock Synchronization

It is important to synchronize the clocks of all workstations so that the log files (which log local system time) will be consistent with each other. JSS Clock Sync is a freeware application (shown in Figure 11) for synchronizing the clock of a PC workstation with a reference network clock. This application was placed on all VBE workstations and run individually on each computer prior to experimentation.

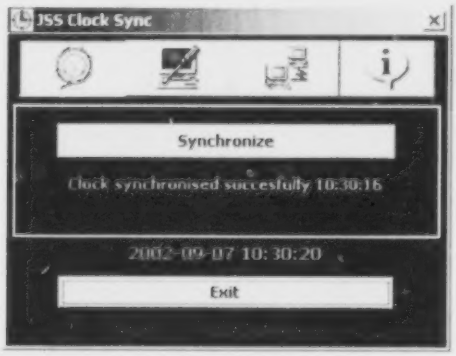

Figure 11. *JSS Clock Synchronization Program*

9.2 Potential Federation Interruptions

When a simulation is run on a shared network rather than a local isolated network, the potential for uncontrollable interruptions is increased. Initial runs of VBE CA-1 were executed on the DRDC Atlantic building network and were disrupted on two occasions. One disruption occurred as a result of an IP number conflict with an incorrectly configured computer elsewhere in the building. The second disruption was the result of a remotely initiated virus scanner that caused the federation to freeze.

Since access to the building network was required only for the synchronization of the workstation clocks, the network cable was unplugged from the wall as soon as the synchronization was complete (prior to execution). This allowed the federation to run

on a local network, isolated from the rest of the building, eliminating unexpected interruptions of the simulation.

9.3 Running the Federation

To run a VBE CA-1 experimental session or demonstration, the following steps must be undertaken:

a. Log on locally to each computer using the user name 'vbe user' and the corresponding password.

b. Click the synchronize button on the JSS Clock Sync window that appears (this step is not necessary if it is just for demonstration purposes).

c. Press Enter on the Run Director keyboard to run the batch file, which starts the RTI and all federates besides the Horizon instances.

d. Press Enter on the other three keyboards to run the remaining batch files (order does not matter).

e. When Horizon has opened in the solo operator position, move the copy of Horizon with blank plug-ins to the screen on the right, maximize it, and click on the 'Ownship' map control (central-right of screen). Maximize the Horizon screen in the left display; click on the TBRG button (bottom-left of the screen) and then on the '30' button (central-left).

f. Repeat step (e) for the coalition operator position, using the left and center display screens.

g. Maximize Horizon in the right screen of the coalition operator position and click on the TBRG button (bottom-left of the screen) and then on the '30' button (central-left).

10. Testing the Simulation

Testing of simulation components and various federation configurations was ongoing throughout the preparation of this experiment. An early federation test revealed a problem with the configuration of Horizon. During this test, two instances of Horizon were running on a single computer, and one instance froze, effectively halting the entire simulation. The error report suggested that Horizon had run out of memory, although this seemed odd as there was ample RAM not being used. Horizon is coded in Java and runs on a Java Virtual Machine (JVM) on its host computer. In this case, both Horizon instances were creating their own JVM. JVM by default uses no more than 75% of the unallocated system memory, so the second copy of JVM was only able to use 75% of the memory unused by the first JVM. Editing the Horizon start-up scripts to specify the limits of its memory use solved this problem.

Once the final federation configuration was identified (as was illustrated in Figure 2), the simulation was set up to run for the entire 2-hour duration of the scenario, and members of the VBE CA-1 team acted as pseudo-operators by making intermittent track associations. Operator interaction was required during the testing phase as it had the potential to affect simulation performance and to reveal any problems with the pop-up query. It was also necessary in order to ensure that the data files were logging associations correctly.

Horizon log files for each operator station were reviewed to ensure all fields were complete and the file format was compared against the expected data format. Sonar log files and pop-up query log files were examined similarly. (Recall that file formats are included in Annex A). Review of the sonar log file indicated that too many tracks were being published, but this turned out to be the result of a configuration error and was simply fixed. All other files were deemed to be correct. (It should be noted, however, that when the analysis of the experimental data was actually completed, it was determined that while associations were being logged correctly, not all associations were being logged. The lesson learned from this was that running test log files through the entire analysis process – actually calculating the metrics – rather than just checking for file recording and format, is an essential step.)

The larger issue was that of obtaining and maintaining real-time performance over the 2-hour scenario. If the scenario ran too slow, the operators would have more time to make their decisions and this would affect the realism of the scenario as well as the results. It would also make it impossible to keep to the planned schedule, where operators completed two 2-hour scenarios and a training session in one day. The details of identifying the source of the slow performance of the simulation and the implemented solution are discussed in detail in [10]. Essentially, through monitoring the performance of various incremental federations, it was identified that federates that were doing the least amount of work were most detrimental to the performance. Following this discovery, a closer look was taken at the sonar federate. In a typical 1000ms time-step, it required less than 100ms to do its works and in the remaining 900ms it polled the RTI 19000 times while awaiting the next time step. Each poll

momentarily used up CPU processing time that could have been used by another federate that still had work to do. Overall, this was slowing down the federation. To deal with this, 50ms sleep cycles were implemented between the polls, reducing the demand on CPU time, and speeding up the federation. With this sleep cycle, the federation as shown in Figure 2 exceeded 99% of real time.

11. Concluding Remarks

The capability to simulate a maritime environment, immerse operators into that environment, and obtain statistically relevant results regarding matters of relevance to the military has been successfully demonstrated through VBE CA-1. This paper has documented the technical issues that were dealt with by the experimental team in order to implement this experiment. The team was left with a greater understanding of the effort that must go into a VBE and can now apply this experience to the planning and execution of future experimentation.

This page intentionally left blank.

12. References

1. Mellema, G.R. and Wentzell, T.E., Experimental Plan for VBE CA-1, DRDC Atlantic Technical Memorandum 2003-158, in review.

2. Mellema, G.R. and Wentzell, T.E., VBE CA-1: An Investigation Into Low-Level Coalition Passive Sonar, DRDC Atlantic Technical Memorandum 2003-223, December 2003.

3. High Level Architecture Federation Development and Execution Process Model, Version 1.5, DMSO, December 8, 1999.

4. Best, J.P., Virtual Ship Architecture Description Document Issue 1.00, Defence Science and Technology Organisation, July 2001.

5. Mellema, G.R. and Wentzell, T.E., Coalition Sonar Track Sharing and VBE CA-1, Proceedings of the Thirteenth Annual TTCP MAR TP-1 Meeting, Dartmouth, NS, Canada, 20-29 October 2003.

6. Arnold, A. and Haddy, M., Horizon 3 - Operator Guide, Release 2.0, Innovation Science Pty Ltd, Australia, December 2001.

7. Cramp, A., Execution Management in the Virtual Ship Architecture Issue 2.00, Defence Science and Technology Organisation, December 2000.

8. Hackett, D., GreenLight Software Solutions, Private Communication, July 2003.

9. Roger, W.A., An Enhanced Scenario Script Generator for the TTCP Virtual Maritime System Architecture, DRDC Atlantic Technical Memorandum 2003-007, February 2003.

10. Mellema, G.R. and Wentzell, T.E., Implementation of VBE CA-1, Proceedings of the Thirteenth Annual TTCP MAR TP-1 Meeting, Dartmouth, NS, Canada, 20-29 October 2003.

This page intentionally left blank.

Annex A: Log File Formats

Table 3 describes the log file format recorded by Horizon.

Table 3. *Horizon Log File Format*

FIELD	TYPE	DESCRIPTION	PRECISION	INVALID VALUE
System Time	String	System Time (hh:mm:ss)		
Data Time	String	Time at which data was recorded (hh:mm:ss)		
Composite Tracks	String	Zero or more strings in format of Track Number field and separated by the '/' character that identify the member tracks of an association		
Track Number	String	Provides unique track identification in the form the <Horizon data source> < track ID>		
Master ID	String	Track master ID as set by the operator, otherwise blank		
Latitude	Real	Latitude in decimal degrees	0.0001	-999.999
Longitude	Real	Longitude in decimal degrees	0.0001	-999.999
Error_Major	Real	Semi-major axis of the ellipse that forms the area of uncertainty	0.1	-1.0
Error_Minor	Real	Semi-Minor axis of the ellipse that forms the area of uncertainty	0.1	-1.0
Error_Angle	Real	Degrees from vertical of the ellipse that forms the area of uncertainty	0.1	-1.0
Course	Real	Degrees[0-360)	0.1	-1.0
Speed	Real	m/s	0.1	-1.0
Source ID	String	Name of the source platform that is providing the track data		
Source Latitude	Real	Latitude in decimal degrees of source platform at data time	0.0001	-999.999
Source Longitude	Real	Longitude in decimal degrees of source platform at data time	0.0001	-999.999
Absolute Bearing	Real	Degrees[0-360) from the source platform to the track	0.1	-1.0

Table 4 describes the log file format recorded by the sonar federate.

Table 4. *Sonar Log File Format*

FIELD	TYPE	DESCRIPTION	PRECISION	APPLICABILITY
Type Code	Integer	101 → initiate new track 201 → update existing track 301 → terminate track		
Track Name	String	Source that provided the track data (e.g., ownship).		
Track Number	Integer	Number assigned to track for identification. Track numbers are unique to the source and assigned sequentially.		
Time Step	Integer	Federation time when track update occurred.		
Bearing	Real	Absolute bearing of the acoustic source(s) (degrees).	0.0001	With Type Codes 101 and 201
Range	Real	Range of the acoustic source (km).	0.0001	With Type Codes 101 and 201
Signal Excess	Real	Signal Excess of the received signal (dB).	0.0001	With Type Codes 101 and 201
Target Count	Integer	Number of targets contributing to this track.		With Type Codes 101 and 201
Target Name(s)	String	Name(s) of targets contributing to this track.		With Type Codes 101 and 201

Annex B: VBE CA-1 Version Fact Sheet

Table 5 shows the version of simulation components that were used for VBE CA-1.

Table 5. *VBE CA-1 Version Fact Sheet*

ITEM	VERSION / DATE
RTI	DMSO RTI 1.3v4
FOM	VMSA 2.0
Horizon	Version 2.1.5 (Update 1 Standard Edition)
Horizon HLA Plug-in	20-Feb-03
Gameboard federate	26-Mar-03
Motion federate	18-Jun-02
Sonar federate	3-Sep-03
VMSEM	18-Jun-02
Enter Reasons Application	28-Aug-03
Coordinate Display Application	11-Aug-03
Bearing Generator Application	Version 1.0
Scenario Generator	Version 4.0.12

This page intentionally left blank.

Annex C: Sonar Configuration File

The sonar configuration file allows the user to select parameters for the sonar federate prior to federation execution. The following is a copy of this file used for VBE CA-1.

```
<SonarConfigurationFile>

        <Environment>
                <BearingVariance>0.1</BearingVariance>
                <FrequencyVariance>5.0</FrequencyVariance>
                <AmbientNoiseMethod>Static</AmbientNoiseMethod>
<!-- values allowed are (Static, Calculate) -->
                <AmbientNoiseLevel>75</AmbientNoiseLevel>
                <ShippingLevel>5</ShippingLevel>    <!-- range is (1-9) -->
                <SeaState>3</SeaState>              <!-- range is (1-7) -->
                <RainState>Intermediate</RainState>
<!-- values allowed are (NoRain, Intermediate, Moderate, Heavy) -->

          <!--Transmission Loss Values allowed are "Simple" (15*log(Range))
              "Thorpe" (calculates absorption loss based on Thorpes algorithim)
    "TableLookup" (looks up transmission loss from the appropriate table)   -->
                <TransmissionLossMethod>Simple</TransmissionLossMethod>
<!-- TransmissionLossTable MinFreq="50" MaxFreq="250" FileName="C:\VBE_CA-
1\VMSA\federates\Sonar\bin\TL_50_250.sap"/ -->
<!-- TransmissionLossTable MinFreq="251" MaxFreq="950" FileName="C:\VBE_CA-
1\VMSA\federates\Sonar\bin\TL_250_950.sap"/    -->
        </Environment>

        <SonarConfiguration>
                <SonarSystem Name="SensorSystem.SonarSystem" Model="Passive"/>
                <DirectivityIndex>19</DirectivityIndex>
                <!-- used to calculate the detection beam angle -->
                <DetectionThreshold>3.0</DetectionThreshold>
                <!-- accuracy used in the signal noise excess calculation -->
                <TimeStep>5</TimeStep>
                <NarrowBand Bandwidth="0.5"/>
                <BroadBand Bandwidth="300" Frequency="150"/>
        </SonarConfiguration>

<!-- "MinRequired/OutOf" specifies how many detections are required before calling a series
of detections a Track -->
<!-- "LoseAfterMissing" specifies how many time cycles need to go by without a detection
before dropping a Track -->
<!-- "DeleteAfter" specifies how many time cycles need to go by with the track set as Lost
before it's removed from the simulation -->
```

```
<TrackerConfiguration>
<NarrowBandDetections MinRequired="15" OutOf="20" LoseAfterMissing="3"
DeleteAfter="30"/>
<BroadBandDetections MinRequired="3" OutOf="5" LoseAfterMissing="5"
DeleteAfter="30"/>
        <GenerateTrackType>BroadBand</GenerateTrackType>
<!-- values allowed are (NarrowBand, BroadBand, Both) -->
        </TrackerConfiguration>

        <TargetProfiles>
                <Target Name="SubSurface.Military.Submarine">
                        <Signal Frequency="200" Level="115" Bandwidth="200"/>
                        <Signal Frequency="400" Level="110" Bandwidth="210"/>
                        <Signal Frequency="60" Level="120" Bandwidth="2"/>
                </Target>

                <Target Name="SeaSurface.NonMilitary.FishingBoat">
                        <Signal Frequency="150" Level="150" Bandwidth="100"/>
                        <Signal Frequency="250" Level="145" Bandwidth="100"/>
                        <Signal Frequency="350" Level="140" Bandwidth="100"/>
                        <Signal Frequency="60" Level="20" Bandwidth="2"/>
                </Target>

                <Target Name="SeaSurface.NonMilitary.Freighter">
                        <Signal Frequency="300" Level="140" Bandwidth="400"/>
                        <Signal Frequency="60" Level="140" Bandwidth="2"/>
                </Target>

                <Target Name="SeaSurface.Military.Warship">
                        <Signal Frequency="300" Level="135" Bandwidth="400"/>
                        <Signal Frequency="60" Level="100" Bandwidth="2"/>
                </Target>

                <Target Name="SeaSurface.NonMilitary.ContainerShip">
                        <Signal Frequency="300" Level="140" Bandwidth="400"/>
                        <Signal Frequency="60" Level="160" Bandwidth="2"/>
                </Target>

                <Target Name="SeaSurface.NonMilitary.Liner">
                        <Signal Frequency="300" Level="140" Bandwidth="400"/>
                        <Signal Frequency="60" Level="180" Bandwidth="2"/>
                </Target>

        </TargetProfiles>

</SonarConfigurationFile>
```

Annex D: Horizon Initialization Files

Horizon initialization files are grouped into categories, denoted by square brackets ([ ]). The categories are collections of fields (to the left of the equal signs (=)) with assigned values (to the right of the equal signs (=)). Number signs (#) at the beginning of a line indicate that the line is a comment.

Filters.ini

```
# Provides default filters that are added to the application upon startup, along
# with mappings from strings to the filters.

[Track Sensor Filters]
# Filters used for track filtering, based on the sensor type allocated for
# the track.
group id = 1
FUSED = track_fused.gif
SONAR = track_ca.gif
# Defines the mappings between a sensor, processor and track type combination,
# and the filter to which the track will be applied.

[Track Filter Map]
* = OTHER
FUSED = FUSED
CA_Sonar = SONAR
AS_Sonar = SONAR
CA_Vessel = SONAR
AS_Vessel = SONAR
```

Fusion.ini

```
# Track Fusion Framework
# Active Fusion Algorithm Definitions

[Combine All]
class = au.com.iscience.navy.horizon.fusion.algs.CombineAllTracks
name = Combine All Tracks
parameters = Hla.ini

[Virtual Ship]
federation name = VBE_CA-1
federate name = Horizon_own
fed file = V:/VBE_CA-1/vmsa/federates/vmsem/bin/vmsfom.fed
som file = V:/VBE_CA-1/vmsa/federates/HorizonHLAPlugin_own/ bin/
HorizonSOM.omt
debug = 1

[Execution Control]
auto connect = true
enable resignation = false
enable iteration termination = true
enable federation termination = true

[Ownship]
# Source from which ownship navigation data is obtained.
# Assumes truth data is always available.
nav source = nav/truth

[GUI]
nav source = nav/truth
execution pane = on
platforms pane = on
nav pane = on
tracks pane = on
annotation pane = off
helm pane = on

[Annotation]
simulation start = on
simulation finish = on
platform added = off
platform updated = off
platform removed = off
report added = off
report updated = off
report removed = off
```

track added = off
track updated = off
track removed = off
associate with track = off

[Helm]
course = on
depth = on
speed = on
course units = degrees
depth units = metres
speed units = knots

[Source Platforms]
valid source = ownship, all

[Tracks]
update = immediate

Map.ini

```
# Parameters for the MapPlugIn

[Initialisation]
# Default lat/long for strait
default latitude = 2.5
default longitude = 101.75

[Visible Ranges]
# The units of measure labels that should be read in by the map.
# Note that this is only a map ini file parameter - it doesn't
# change the default units of measure used by Horizon.
# Valid values are "km" for kilometres or "NM" for nautical miles
# (case is important).
units = km

# The maximum range (in the same units of measure as specified above)
# that the map will search for shoreline tiles.
maximum range = 1000

# The five values for the visible range selection buttons when the
# map is displaying ranges in kilometres.
km1 = 5
km2 = 10
km3 = 20
km4 = 50
km5 = 100

# The five values for the visible range selection buttons when the
# map is displaying ranges in nautical miles.
NM1 = 5
NM2 = 10
NM3 = 20
NM4 = 50
NM5 = 100

[Shoreline]
# The expected size of a shoreline data tile, in degrees of latitude and longitude.
Tile size = 4
#  Tile data path should be relative to ini file path.
tile data path = ../map_tiles

[Track History]
# The period (in seconds) between track history trail markers being "dropped"
# onto the screen.
drop threshold = 60
```

```
# The search threshold is the maximum amount of track history data (in seconds)
# that will be searched when a valid track lat/long position is required
# Warning: reducing this value will cause tracks that don't have any new data
# will be removed from the display if their most recent record is still beyond
# the bounds of the threshold.
search threshold = 5400

[Range Rings]
# Indicates if range rings should be drawn anti-aliased or not.  Anti-aliased
# rings look better, but take longer to draw.
anti-alias = true

[Error Ellipses]
# The period (in seconds) between track error ellipse markers being "dropped"
# onto the screen.
drop threshold = 60

# Indicates if error ellipses should be drawn anti-aliased or not.  Anti-aliased
# ellipses look better, but take longer to draw.
anti-alias = true

# The maximum number of error ellipses that can be displayed before they will
# automatically be drawn without anti-aliasing (for efficiency reasons).
anti-alias threshold = 60

[Rangeless Tracks]
# The length, in pixels, of the tag that connects a rangeless track marker to
# the outer range ring.
tag length = 10

# The distance, in pixels, from the outer range ring to the centre of a rangeless
# track marker.
marker distance = 20

[Bearing Fans]
# The period (in seconds) between track bearing fans being "dropped" onto the screen.
drop threshold = 30
# Indicates if bearing fans should be drawn anti-aliased or not.  Anti-aliased
# fans look better, but take longer to draw.
anti-alias = true

# Indicates if locality rings should be drawn anti-aliased or not.  Anti-aliased
# rings look better, but take longer to draw.
locality rings anti-alias = true

# The size (in pixels) of a locality ring.
locality rings size = 200
```

```
[Dead Reckoning]
# Threshold used to determine when an indication that track data is not being
# dead reckoned should be displayed.  For a track that is not being dead
# reckoned, if the time of the track's latest system solution is older than
# the threshold, the track will be displayed with a cross behind it to
# indicate the data is not up to date.  The parameter value is in seconds.
 non dr threshold = 5
```

Plugin.ini

```
[Map Display]
class = au.com.iscience.navy.trackfusion.map.MapPlugIn
default location = PLUGIN_MAIN
displayable = true
standard socket compatible = true
main socket compatible = true
auto create = true

[Time/Bearing]
class = au.com.iscience.navy.trackfusion.timebearing.TimeBearingPlugIn
displayable = true
main socket compatible = true

[System Roles]
class = au.com.iscience.navy.trackfusion.system.SystemRolePlugIn
default location = PLUGIN_SYSTEM
displayable = true
system socket compatible = true
number of instances = 1

[Recorder]
class = au.com.iscience.navy.trackfusion.recorder.RecorderPlugIn
displayable = false
number of instances = 1
default params = c:/binary_log_files

[VBERecorder]
class = au.com.iscience.navy.trackfusion.recorder.VBE_RecorderPlugIn
displayable = false
number of instances = 1
default params = /text_log_files

[Active Fusion]
class = au.com.iscience.navy.horizon.fusion.ActiveFusionPlugIn
default location = PLUGIN_1
displayable = true
standard socket compatible = true
number of instances = 1

[Annotation Generator]
class = au.com.iscience.navy.trackfusion.generators.AnnotationGenerator
default location = PLUGIN_2
displayable = true
standard socket compatible = true
number of instances = 1
```

```
[Datum]
class = au.com.iscience.navy.horizon.generators.DatumPlugIn
displayable = true
standard socket compatible = true
default params = D
number of instances = 1

[Cursor Position]
class = au.com.iscience.navy.trackfusion.position.CursorPositionPlugIn
default location = PLUGIN_4
displayable = true
standard socket compatible = true
number of instances = 1

[Annotation Maintenance]
class = au.com.iscience.navy.trackfusion.maint.AnnotationMaintenancePlugIn
default location = PLUGIN_3
displayable = true
standard socket compatible = true
number of instances = 1

[Horizon HLA Plugin]
class = au.com.iscience.navy.trackfusion.federate.HorizonHLAPlugIn
displayable = true
standard socket compatible = false
main socket compatible = true
number of instances = 1
```

Recorder.ini

```
[General]
# If true, indicates that all track history records that are marked as
# system solutions will be recorded. Consult the "Solution Data" section
# in the javadocs of the "Track" class for more information on solution types.
record system solution = true

# If true, indicates that the record should record all track history records
# that are marked as "raw" solutions. Consult the "Solution Data" section
# in the javadocs of the "Track" class for more information on solution types.
record raw solution = true

# If true, indicates that the record should record all track history records
# that are marked as "trial" solutions. Consult the "Solution Data" section
# in the javadocs of the "Track" class for more information on solution types.
record trial solution = true

# Tracks can be tagged as "automatically generated" if they are created as a
# result of a track "fusion". Tracks received from external sources are not
# tagged as "automatically generated" (at least for existing communications
# plugins). Setting this parameter to true will cause all tracks to be recorded
# regardless of their "automatically generated" state. Setting this parameter
# to false will cause only non-automatically generated tracks to be recorded.
record automatically generated = true
```

Roles.ini

```
[Initialisation]
title string = Menu
label 1 = HORIZON 3
label 2 = v2.1.5
read only = true

[PRESET 1]
title = MAP
PLUGIN_MAIN = 1,Map Display

[PRESET 2]
title = TBRG
PLUGIN_MAIN = 1,Time/Bearing

[PRESET 3]
title = VBE
PLUGIN_1 = 1,Active Fusion
PLUGIN_2 = 1,Annotation Generator
PLUGIN_3 = 1,Annotation Maintenance
PLUGIN_4 = 1,Cursor Position

[PRESET 4]
title = more...
PLUGIN_1 = 1,Active Fusion
PLUGIN_2 = 1,Annotation Generator
PLUGIN_3 = 1,Cursor Position
PLUGIN_4 = 1,Datum
```

TrackFusion.ini

```
# Application wide parameters for Track Fusion.

[Configuration]
configuration name = Standard Edition

[Repository]
auto update external parameters = true

# Indicates if dead reckoning should be applied to tracks that have no
# incoming data. The "dead reckon threshold" is the time (in seconds)
# for which a track must receive no data before it will be dead reckoned
dead reckon = true
dead reckon threshold = 10

# Application wide range of the day (in metres), used for source platforms
# that haven't had a specific range of the day set.
range of the day = 10000

# If true, instructs the framework to automatically delete tracks
# that have been dropped.
auto delete dropped tracks = false

[State Data]
# Indicates if state data should automatically deselect objects when changing
# to a selection mode with a different object type.
automatic deselection = true

[Framework]
disable socket menu = false

[Alerts]
# This section defines the way alert coalescence works.
# Parameters:
coalescence enabled = true
coalescence ignore type = false
coalescence ignore text = false
coalescence ignore object = true
coalescence delete existing = true

[Units of Measure]
# Defines the default units of measure to use for standard
# display axes.

True Bearing = DEGREES
Relative Bearing = DEGREES
```

Time = SECONDS
Range = METRES
Depth = METRES
Speed = KNOTS
Course = DEGREES
Heading = DEGREES
Angle to Bow = DEGREES
Temperature = DEGRESS CELCIUS
Latitude = DEGREES
Longitude = DEGREES

VbeRecorder.ini

```
[General]
# Time between output of log data (in seconds).
delay period = 15.0

# String used to separate fields in the log file.
field separator = ,

# String used to separate multiple track numbers.
track separator = /

# Indicates if automatically generated (fused) tracks should be recorded.
record automatically generated = true

[Logger]
# Logging parameters.  The first parameter turns logging on and off without
# requiring adjustment to the plugin.ini file.
logging on = true

# Directory in which the log file will be placed.
#log path = /tmp
 log path = c:/text_log_files

# Filename prefix for the log file.
log filename = horizon_data

# Filename suffix for the log file.
log extension = .log

# When set to true, the log file will use the current date/time in its filename,
# down to the current minute.  Otherwise, it will overwrite the previous log file
# each time Horizon is restarted.
log unique filename = true
```

This page intentionally left blank.

Annex E: Gameboard Scenario Control File for Training Scenario

The Gameboard scenario control file defines platforms and their kinematic details throughout a scenario. The following file was used for VBE CA-1.

```
<gc:Scenario StartTime="00:00:00" StopTime="02:00:00" xmlns:gc="http://www.drdc-
rddc.gc.ca/Namespace/drdca/vmsa/gc" xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xsi:schemaLocation="http://www.drdc-rddc.gc.ca/Namespace/drdca/vmsa/gc
.\..\Gameboard.xsd">
 <gc:AddVessel Affiliation="Blue" Name="CA_Vessel" Time="00:00:00"
Type="SeaSurface.Military.Warship.Frigate.Canada...CA_Vessel">
  <gc:Position Latitude="2.47788739869363" Longitude="101.716636321791"/>
  <gc:Speed Value="12" Units="KNOTS"/>
  <gc:Course Value="94.87399"/>
  <gc:Depth Value="0" Units="METRES"/>
 </gc:AddVessel>
 <gc:AddVessel Affiliation="Blue" Name="AS_Vessel" Time="00:00:00"
Type="SeaSurface.Military.Warship.Frigate.Australia...AS_Vessel">
  <gc:Position Latitude="2.17803446054885" Longitude="101.533590300532"/>
  <gc:Speed Value="12" Units="KNOTS"/>
  <gc:Course Value="120.4688"/>
  <gc:Depth Value="0" Units="METRES"/>
 </gc:AddVessel>
 <gc:AddVessel Affiliation="Blue" Name="Freight_15S1" Time="00:00:00"
Type="SeaSurface.NonMilitary.Freighter.Large.NewZealand...Freight_15S1">
  <gc:Position Latitude="2.39114840640105" Longitude="101.522267592732"/>
  <gc:Speed Value="15" Units="KNOTS"/>
  <gc:Course Value="124.6913"/>
  <gc:Depth Value="0" Units="METRES"/>
 </gc:AddVessel>
 <gc:AddVessel Affiliation="Blue" Name="Freight_15N1" Time="00:00:00"
Type="SeaSurface.NonMilitary.Freighter.Large.NewZealand...Freight_15N1">
  <gc:Position Latitude="2.09874596844652" Longitude="102.003477462151"/>
  <gc:Speed Value="15" Units="KNOTS"/>
  <gc:Course Value="305.8624"/>
  <gc:Depth Value="0" Units="METRES"/>
 </gc:AddVessel>
 <gc:AddVessel Affiliation="Blue" Name="Freight_20S1" Time="00:00:00"
Type="SeaSurface.NonMilitary.Freighter.Large.NewZealand...Freight_20S1">
  <gc:Position Latitude="2.47411113516225" Longitude="101.255010307845"/>
  <gc:Speed Value="20" Units="KNOTS"/>
  <gc:Course Value="122.6142"/>
  <gc:Depth Value="0" Units="METRES"/>
 </gc:AddVessel>
```

```
 <gc:AddVessel Affiliation="Blue" Name="Freight_20S2" Time="00:00:00"
Type="SeaSurface.NonMilitary.Freighter.Large.NewZealand...Freight_20S2">
  <gc:Position Latitude="2.29024653284324" Longitude="101.597043564844"/>
  <gc:Speed Value="20" Units="KNOTS"/>
  <gc:Course Value="123.6841"/>
  <gc:Depth Value="0" Units="METRES"/>
 </gc:AddVessel>
 <gc:AddVessel Affiliation="Blue" Name="Freight_20N1" Time="00:00:00"
Type="SeaSurface.NonMilitary.Freighter.Large.NewZealand...Freight_20N1">
  <gc:Position Latitude="1.85954824027582" Longitude="102.17652389265"/>
  <gc:Speed Value="20" Units="KNOTS"/>
  <gc:Course Value="303.1238"/>
  <gc:Depth Value="0" Units="METRES"/>
 </gc:AddVessel>
 <gc:AddVessel Affiliation="Blue" Name="Freight_8S1" Time="00:00:00"
Type="SeaSurface.NonMilitary.Freighter.Large.NewZealand...Freight_8S1">
  <gc:Position Latitude="2.55341649270784" Longitude="101.524796076027"/>
  <gc:Speed Value="8" Units="KNOTS"/>
  <gc:Course Value="123.3454"/>
  <gc:Depth Value="0" Units="METRES"/>
 </gc:AddVessel>
 <gc:AddVessel Affiliation="Blue" Name="Freight_8S2" Time="00:00:00"
Type="SeaSurface.NonMilitary.Freighter.Large.NewZealand...Freight_8S2">
  <gc:Position Latitude="2.41912695797209" Longitude="101.668214394956"/>
  <gc:Speed Value="8" Units="KNOTS"/>
  <gc:Course Value="127.0039"/>
  <gc:Depth Value="0" Units="METRES"/>
 </gc:AddVessel>
 <gc:AddVessel Affiliation="Blue" Name="Freight_8S3" Time="00:00:00"
Type="SeaSurface.NonMilitary.Freighter.Large.NewZealand...Freight_8S3">
  <gc:Position Latitude="2.32705890624282" Longitude="101.862962421687"/>
  <gc:Speed Value="8" Units="KNOTS"/>
  <gc:Course Value="126.2966"/>
  <gc:Depth Value="0" Units="METRES"/>
 </gc:AddVessel>
 <gc:ModifyVessel Name="CA_Vessel" Time="00:25:41">
  <gc:Course Value="167.4742"/>
 </gc:ModifyVessel>
 <gc:ModifyVessel Name="CA_Vessel" Time="00:51:47">
  <gc:Course Value="95.96009"/>
 </gc:ModifyVessel>
 <gc:ModifyVessel Name="CA_Vessel" Time="01:19:11">
  <gc:Course Value="143.6836"/>
 </gc:ModifyVessel>
 <gc:ModifyVessel Name="CA_Vessel" Time="01:43:02">
  <gc:Course Value="88.46836"/>
 </gc:ModifyVessel>
 <gc:ModifyVessel Name="CA_Vessel" Time="02:00:00">
```

```
    <gc:Remove/>
  </gc:ModifyVessel>
  <gc:ModifyVessel Name="AS_Vessel" Time="00:21:26">
    <gc:Course Value="73.85715"/>
  </gc:ModifyVessel>
  <gc:ModifyVessel Name="AS_Vessel" Time="00:43:49">
    <gc:Course Value="120.148"/>
  </gc:ModifyVessel>
  <gc:ModifyVessel Name="AS_Vessel" Time="01:04:08">
    <gc:Course Value="157.9339"/>
  </gc:ModifyVessel>
  <gc:ModifyVessel Name="AS_Vessel" Time="01:26:40">
    <gc:Course Value="105.9541"/>
  </gc:ModifyVessel>
  <gc:ModifyVessel Name="AS_Vessel" Time="01:47:15">
    <gc:Course Value="163.8559"/>
  </gc:ModifyVessel>
  <gc:ModifyVessel Name="AS_Vessel" Time="02:00:00">
    <gc:Remove/>
  </gc:ModifyVessel>
  <gc:ModifyVessel Name="Freight_15S1" Time="02:00:00">
    <gc:Remove/>
  </gc:ModifyVessel>
  <gc:ModifyVessel Name="Freight_15N1" Time="02:00:00">
    <gc:Remove/>
  </gc:ModifyVessel>
  <gc:ModifyVessel Name="Freight_20S1" Time="02:00:00">
    <gc:Remove/>
  </gc:ModifyVessel>
  <gc:ModifyVessel Name="Freight_20S2" Time="02:00:00">
    <gc:Remove/>
  </gc:ModifyVessel>
  <gc:ModifyVessel Name="Freight_20N1" Time="02:00:00">
    <gc:Remove/>
  </gc:ModifyVessel>
  <gc:ModifyVessel Name="Freight_8S1" Time="02:00:00">
    <gc:Remove/>
  </gc:ModifyVessel>
  <gc:ModifyVessel Name="Freight_8S2" Time="02:00:00">
    <gc:Remove/>
  </gc:ModifyVessel>
  <gc:ModifyVessel Name="Freight_8S3" Time="02:00:00">
    <gc:Remove/>
  </gc:ModifyVessel>
</gc:Scenario>
```

This page intentionally left blank.

Annex F: VBE CA-1 Scenario Components File

The scenario components file indicates what federates are participating in the federation and what components (sonar, Horizon displays, etc.) will exist on particular platforms. The following such file was used for VBE CA-1.

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- <!DOCTYPE ScriptFile PUBLIC "IUW Script file" "file:///V:/VBE_CA-
1/vmsa/federates/vmsem/doc/ScriptFile.dtd"> -->
<ScriptFile>
<Header RealTime="5.0" FederationName="VBE_CA-1">
<ParticipatingFederate Name="Motion"/>
<ParticipatingFederate Name="Horizon_allied"/>
<ParticipatingFederate Name="Horizon_own"/>
<ParticipatingFederate Name="Horizon_own2"/>
<ParticipatingFederate Name="Horizon_own3"/>
<ParticipatingFederate Name="Horizon_own4"/>
<ParticipatingFederate Name="Gameboard"/>
<ParticipatingFederate Name="Sonar"/>

</Header>
<Simulation>
<Loop MaxLoops="100">
<ScenarioDescription>VBE CA-1 Practice Scenario</ScenarioDescription>
<RandomNumberSeed>loop_index</RandomNumberSeed>

<CompositeEntity Type="SeaSurface" Time="0.0" Name="AS_Vessel"
FederateName="Gameboard" ConfigFile="Gameboard.cfg">

<ComponentEntity Type="CommandAndControlSystem" Name="AS_Horizon"
FederateName="Horizon_allied" ConfigFile="horizon.cfg">
<ComponentEntityAttribute AttributeName="ComponentName">
<AttributeEntry AttributeValue="CommandAndControlSystem...." AttributeType="s"/>
</ComponentEntityAttribute>
</ComponentEntity>

<ComponentEntity Type="SensorSystem.SonarSystem" Name="AS_Sonar"
FederateName="Sonar" ConfigFile="sonar.cfg">
<ComponentEntityAttribute AttributeName="ComponentName">
<AttributeEntry AttributeValue="SensorSystem.SonarSystem..." AttributeType="s"/>
</ComponentEntityAttribute>
</ComponentEntity>

</CompositeEntity>
```

```
<CompositeEntity Type="SeaSurface" Time="0.0" Name="CA_Vessel"
FederateName="Gameboard" ConfigFile="Gameboard.cfg">

<ComponentEntity Type="CommandAndControlSystem" Name="CA_Horizon"
FederateName="Horizon_own" ConfigFile="horizon.cfg">
<ComponentEntityAttribute AttributeName="ComponentName">
<AttributeEntry AttributeValue="CommandAndControlSystem...." AttributeType="s"/>
</ComponentEntityAttribute>
</ComponentEntity>

<ComponentEntity Type="CommandAndControlSystem" Name="CA_Horizon2"
FederateName="Horizon_own2" ConfigFile="horizon.cfg">
<ComponentEntityAttribute AttributeName="ComponentName">
<AttributeEntry AttributeValue="CommandAndControlSystem...." AttributeType="s"/>
</ComponentEntityAttribute>
</ComponentEntity>

<ComponentEntity Type="CommandAndControlSystem" Name="CA_Horizon3"
FederateName="Horizon_own3" ConfigFile="horizon.cfg">
<ComponentEntityAttribute AttributeName="ComponentName">
<AttributeEntry AttributeValue="CommandAndControlSystem...." AttributeType="s"/>
</ComponentEntityAttribute>
</ComponentEntity>

<ComponentEntity Type="CommandAndControlSystem" Name="CA_Horizon4"
FederateName="Horizon_own4" ConfigFile="horizon.cfg">
<ComponentEntityAttribute AttributeName="ComponentName">
<AttributeEntry AttributeValue="CommandAndControlSystem...." AttributeType="s"/>
</ComponentEntityAttribute>
</ComponentEntity>

<ComponentEntity Type="SensorSystem.SonarSystem" Name="CA_Sonar"
FederateName="Sonar" ConfigFile="sonar.cfg">
<ComponentEntityAttribute AttributeName="ComponentName">
<AttributeEntry AttributeValue="SensorSystem.SonarSystem..." AttributeType="s"/>
</ComponentEntityAttribute>
</ComponentEntity>

</CompositeEntity>

<EndTime>12000</EndTime>
</Loop>
</Simulation>
</ScriptFile>
```

List of symbols/abbreviations/acronyms/initialisms

AAW	Anti-Air Warfare
CoaSTS	Coalition Sonar Track Sharing
DMSO	Defence Modeling and Simulation Office
DRDC Atlantic	Defence Research and Development Canada – Atlantic
DSTO	Defence Science and Technology Organization
FOM	Federation Object Model
HIL	Human-in-the-Loop
HLA	High Level Architecture
JVM	Java Virtual Machine
MICS	Maritime Information and Combat Systems
RTI	Run Time Infrastructure
SOM	Simulation Object Model
TMA	Target Motion Analysis
TTCP MAR TP-1	The Technical Cooperation Program Maritime Technical Panel-1
UDF	Underwater Data Fusion
VB	Visual Basic
VBE CA-1	Virtual Battle Experiment Canada-1
VCS	Virtual Combat Systems
VMSEM	Virtual Maritime System Execution Management
VMSSD	Virtual Maritime System Situational Display

This page intentionally left blank.

Distribution List

LIST PART 1: CONTROLLED BY DRDC ATLANTIC LIBRARY

2 DRDC ATLANTIC LIBRARY FILE COPIES
3 DRDC ATLANTIC LIBRARY (SPARES)

1 J. S. Kennedy
1 M. G. Hazen
1 W. A. Roger
1 B. A. McArthur
1 T. E. Wentzell
1 G. R. Mellema
1 J. Murphy
1 B. Dillman
1 A. Gillis
1 M. E. Lefrançois
1 A. W. Isenor
1 K. McTaggert
1 D. Heath
1 D. Perrault

19 TOTAL LIST PART 1

LIST PART 2: DISTRIBUTED BY AUTHOR

1 Eric Chaum
TTCP MAR TP-1 USNL
Naval Undersea Warfare Center
Division Newport
Code 2231, Bldg 1171-3
1176 Howell Street
Newport, RI 02841-1708

1 Richard King
TTCP MAR TP-1 Chair
Naval Systems Department, C-134
Defence Science and Technology Laboratory
Portsdown West, Fareham, Hampshire
PO17 6AD, United Kingdom

1 Graham Macferson
TTCP MAR TP-1 NZNL
Section Manager, C4I & Simulation
Defence Technology Agency
Private Bag 32901
Auckland, New Zealand

1 Dr. John Riley
TTCP MAR TP-1 ASNL
Defence Science and Technology Organisation
Maritime Operations Division
PO Box 1500, Edinburgh, SA 5111
Australia

1 Robert Walden
TTCP MAR TP-1 UKNL
Naval Systems Department, C-134/EC
Defence Science and Technology Laboratory
Portsdown West, Fareham, Hampshire
P017 6AD, United Kingdom

5 TOTAL LIST PART 2

LIST PART 3: DISTRIBUTED BY DRDKIM 3

1 NDHQ/ ADMS&T/ DRDKIM 3

1 TOTAL LIST PART 3

25 TOTAL COPIES REQUIRED

DOCUMENT CONTROL DATA

(Security classification of title, body of abstract and indexing annotation must be entered when the overall document is classified)

1. ORIGINATOR (the name and address of the organization preparing the document. Organizations for whom the document was prepared, e.g. Centre sponsoring a contractor's report, or tasking agency, are entered in section 8.)

Defence R&D Canada – Atlantic
PO Box 1012
Dartmouth, Nova Scotia
B2Y 3Z7

2. SECURITY CLASSIFICATION (overall security classification of the document including special warning terms if applicable).

UNCLASSIFIED

3. TITLE (the complete document title as indicated on the title page. Its classification should be indicated by the appropriate abbreviation (S,C,R or U) in parentheses after the title).

Technical Preparation for an Initial Virtual Battle Experiment

4. AUTHORS (Last name, first name, middle initial. If military, show rank, e.g. Doe, Maj. John E.)

Tania E. Wentzell and Garfield R. Mellema

5. DATE OF PUBLICATION (month and year of publication of document)	6a. NO. OF PAGES (total containing information Include Annexes, Appendices, etc).	6b. NO. OF REFS (total cited in document)
July 2004	66	10

7. DESCRIPTIVE NOTES (the category of the document, e.g. technical report, technical note or memorandum. If appropriate, enter the type of report, e.g. interim, progress, summary, annual or final. Give the inclusive dates when a specific reporting period is covered).

DRDC Atlantic TM 2004-014

8. SPONSORING ACTIVITY (the name of the department project office or laboratory sponsoring the research and development. Include address).

Defence R&D Canada – Atlantic
PO Box 1012
Dartmouth, NS, Canada B2Y 3Z7

9a. PROJECT OR GRANT NO. (if appropriate, the applicable research and development project or grant number under which the document was written. Please specify whether project or grant).	9b. CONTRACT NO. (if appropriate, the applicable number under which the document was written).

10a ORIGINATOR'S DOCUMENT NUMBER (the official document number by which the document is identified by the originating activity. This number must be unique to this document.)	10b OTHER DOCUMENT NOs. (Any other numbers which may be assigned this document either by the originator or by the sponsor.)
DRDC Atlantic TM 2004-014	

11. DOCUMENT AVAILABILITY (any limitations on further dissemination of the document, other than those imposed by security classification)

(x) Unlimited distribution
( ) Defence departments and defence contractors; further distribution only as approved
( ) Defence departments and Canadian defence contractors; further distribution only as approved
( ) Government departments and agencies; further distribution only as approved
( ) Defence departments; further distribution only as approved
( ) Other (please specify):

12. DOCUMENT ANNOUNCEMENT (any limitation to the bibliographic announcement of this document. This will normally correspond to the Document Availability (11). However, where further distribution (beyond the audience specified in (11) is possible, a wider announcement audience may be selected).

13. ABSTRACT (a brief and factual summary of the document. It may also appear elsewhere in the body of the document itself. It is highly desirable that the abstract of classified documents be unclassified. Each paragraph of the abstract shall begin with an indication of the security classification of the information in the paragraph (unless the document itself is unclassified) represented as (S), (C), (R), or (U). It is not necessary to include here abstracts in both official languages unless the text is bilingual).

The Virtual Combat Systems (VCS) Group at Defence Research and Development Canada Atlantic (DRDC Atlantic) is aiming to provide a combat systems experimentation facility to assist with the research programs of scientists within the Maritime Information and Combat Systems (MICS) section and beyond. The first such experiment, known as Virtual Battle Experiment Canada-1 (VBE CA-1), investigated the influence of coalition sharing of broadband passive sonar track segment data.

The experiment required sonar operators to monitor vessel traffic in a strait and make decisions about which sonar tracks were from common targets. In some cases, sonar track data was available to the operator from a coalition ship, and in other cases the operators were only provided with sonar track data from their own sonar. The experiment was run multiple times with various subjects to obtain statistically relevant results.

This paper focuses on the technical activities that were completed in preparation for this experiment. Scenario development, selection, development and integration of simulation components, and testing of the architecture are discussed.

14. KEYWORDS, DESCRIPTORS or IDENTIFIERS (technically meaningful terms or short phrases that characterize a document and could be helpful in cataloguing the document. They should be selected so that no security classification is required. Identifiers, such as equipment model designation, trade name, military project code name, geographic location may also be included. If possible keywords should be selected from a published thesaurus, e.g. Thesaurus of Engineering and Scientific Terms (TEST) and that thesaurus-identified. If it not possible to select indexing terms which are Unclassified, the classification of each should be indicated as with the title).

Virtual Battle Experiment
VBE
VBE CA-1
Virtual Experimentation
High Level Architecture
HLA
Virtual Maritime System Architecture
VMSA
Human-in-the-Loop
HIL